Resource constrained project scheduling models and algorithms applied to underground mining

SE Terblanche

Dissertation presented for the degree of
Doctor of Philosophy
in the Faculty of Engineering at Stellenbosch University

Promoter: Prof JH van Vuuren

December 2017

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December 2017

Abstract

The *resource constrained project scheduling problem* (RCSP) involves the scheduling of a number of activities over time, where each activity consumes a unit of some resource per time period. For a feasible solution to exist, the total resource consumption per time period must be less than or equal to the available resources. In addition, the order in which activities may be scheduled is determined by a precedence graph. The nodes of this directed graph represent the various activities and each directed edge a precedence relationship. A multitude of model formulations for the RCSP exist in the literature as do various solution approaches. Some of the frequently applied objective functions include the minimisation of the makespan, the minimisation of a tardiness penalty cost, and the maximisation of net present value.

The advent of practical computer technology during the late 1950s has meant that various industrial problems can now be solved by computer algorithms. It soon became clear, however, that certain types of problems are inherently difficult and in some cases even impossible to solve. Even today scheduling problems exist which have no more than 60 tasks to be scheduled, but which cannot be solved to optimality within reasonable time using the latest algorithmic and computer technology.

In this dissertation, the challenges of underground mine planning are addressed by employing RCSP models and algorithms. Underground mine planning entails the scheduling of mining activities in a manner that the most economical value is derived, while satisfying constraints related to resource requirements and physical limitations due to the properties of the mine infrastructure. Modelling extensions that address mining-specific requirements are of specific interest. For instance, the modelling of transfer delay constraints are especially useful in a mechanised mining environment where the movement of large machinery from one point to another may cause significant delays in a mine production schedule. Other mining-specific requirements include the modelling of uncertainty in resource requirements as well as the formulation of scheduling models that facilitate selective scheduling.

Details of existing RCSP formulations in the literature are provided and results from empirical tests are presented to evaluate the suitability of adopting a resource flow-based RCSP formulation to solve the underground mine scheduling optimisation problem. Modifications to the resource flow formulation are proposed for the purpose of accommodating the maximisation of net present value. Due to the computational complexity of the underground mine scheduling problem, variable and constraint reduction approaches are suggested. In addition, a Benders decomposition approach is described which is capable of improving the computation of feasible solutions for large problem instances.

Computational results presented in this dissertation are based on both randomly generated data and data from a real South African underground mine. Based on these results it is found that the best performing model reformulation involves the use of a resource flow-based model in conjunction with a constraint aggregation and graph reduction approach. The Benders decomposition approach, implemented within a branch-and-cut framework, scales well for problem instances with a large number of activities and resources. This is a significant contribution within the context of mining, especially considering the large number of resources that have to be accommodated when solving underground mine scheduling optimisation problems.

Uittreksel

Die *hulpbronbeperkte skeduleringsprobleem* (HBSP) behels die skedulering van 'n aantal aktiwiteite oor verloop van tyd, waar elke aktiwiteit 'n eenheid van 'n sekere hulpbron per tydeenheid verbruik. Vir 'n haalbare oplossing om te bestaan, moet die totale hulpbron verbruik per tydeenheid hoogstens die beskikbare hulpbronne wees. Die volgorde waarin aktiwiteite geskeduleer kan word, word verder deur 'n voorrang-grafiek bepaal. Die punte van hierdie gerigte grafiek verteenwoordig die verskillende aktiwiteite en elke boog dui op 'n voorrang verhouding. Modelformulerings vir verskeie variasies op die HBSP kan in die literatuur gevind word, asook 'n verskeidenheid oplosmetodes. Van die mees gewilde doelfunksies behels die minimering van die totale projekduur, die minimering van 'n traagheids-strafkoste, en die maksimering van die netto teenswoordige waarde.

Die ontwikkeling van praktiese rekenaartegnologie in die laat 1950s het beteken dat verskeie bedryfsprobleme nou ook deur middel van rekenaaralgoritmes opgelos kan word. Dit het egter gou duidelik geword dat sekere tipes probleme inherent moeilik is en dat dit in sommige gevalle selfs onmoontlik is om hierdie probleme op te los. Tot op hede bestaan daar selfs skeduleringsprobleme wat nie meer as 60 aktiwiteite het nie, en wat nie optimaal binne 'n redelike tydsverloop opgelos kan word nie, selfs al word daar van die nuutste rekenaartegnologie en algoritmes gebruik gemaak.

In hierdie proefskrif word die uitdagings van ondergrondse mynboubeplanning aangespreek deur gebruik te maak van HBSP modelle en algoritmes. Ondergrondse mynboubeplanning behels die skedulering van mynboubedrywighede op só 'n wyse dat die mees ekonomiese waarde daaruit geput kan word. Dit moet gedoen word in aggenome sekere vereistes ten opsigte van hulpbronbeperkings, asook beperkings wat daar gestel word deur die fisiese infrastruktuur van die myn. Van spesifieke belang is die uitbreiding van bestaande HBSP modelle om mynbou-spesifieke vereistes aan te spreek. 'n Voorbeeld hiervan is die modellering van toelaatbare vertragingstyd, vir wanneer groot masjinerie in 'n gemeganiseerde mynbou omgewing van een punt na 'n ander vervoer moet word.

Besonderhede van reeds bestaande HBSP modelle word in hierdie proefskrif voorgehou. Empiriese toetse word gebruik om die toepaslikheid van hulpbron vloei-gebaseerde formulerings vir die oplos van die mynbouskeduleringsprobleem te staaf. Aanpassings word vir die hulpbron vloei-gebaseerde formulering voorgestel om sodoende die maksimering van die netto huidige waarde te akkommodeer. Verskeie herformulerings word voorgestel as gevolg van die noemenswaardige berekeningskompleksiteit van die ondergrondse mynbouskeduleringsprobleem. Verder word 'n Benders ontbindingsbenadering beskryf wat in staat is om die berekeningstyd van oplossings vir groot probleemgevalle te verminder.

Resultate wat in hierdie proefskrif gerapporteer word, is gebaseer op beide lukraak-gegenereerde data en data wat afkomstig is van 'n Suid-Afrikaanse ondergrondse myn. Die bevindings van hierdie studie is dat die model wat die beste vaar, gebruik maak van 'n hulpbron vloei-gebaseerde formulering, in samewerking met 'n formuleringsbenadering wat 'n verminderde aantal veranderlikes en beperkings tot gevolg het. Die resultate vir die Benders ontbindingsbenadering wys daarop dat die metode skaalbaar is vir probleemgevalle met 'n groot aantal aktiwiteite en hulpbronne. Dit is 'n belangrike bydrae in die konteks van mynbou, veral in ag geneem dat 'n groot aantal hulpbronne in die oplossing van ondergrondse mynbouskeduleringsprobleme beskou moet word.

Acknowledgements

The author wishes to acknowledge the following people and institutions for their various contributions towards the completion of this work:

- To Prof Jan van Vuuren, thank you for your guidance and assistance with the manuscript.
- To MineRP, for providing me with access to the necessary planning tools and data.
- To Mariet, my lovely wife. Thank you for your unconditional love and support.

Table of Contents

List of Acronyms

AC: Aggregated constraints

CAD: Computer aided design

EB: Event-based

ECS: Explicit cutset inequalities

GDP: Gross domestic product

GR: Graph reduction

ICS: Implicit cutset inequalities

ILP: Integer linear programming

LP: Linear programming

MILP: Mixed integer linear programming

NPV: Net present value

PSPLIB: Project scheduling problem library

RCSP: Resource constrained scheduling problem

RF: Resource flow

RFM: Resource flow Benders master problem

RFSEP: Resource flow separation problem

RFSEP-AC: Resource flow separation problem with aggregated constraints

RFSEP-DA: Resource flow separation problem with disaggregated constraints

RFSEP-SLP: Resource flow separation problem with a single LP implementation

RFSEP-PDA: A parallel implementation of the RFSEP-DA

RSPH: Right-shift primal heuristic

TI: Time-indexed

TRSH: Time-indexed right-shift heuristic

List of Figures

List of Tables

CHAPTER 1

Introduction

Contents

> “*A change so unexpected and a development never known before were due to the discovery in 1886 of the greatest gold mines of all history, ancient and modern. From 1886 the story of South Africa is the story of gold.*”
>
> C.W. de Kiewiet, *South Africa*, 1941

Mining in South Africa has its origins in the discovery of the first diamond on the banks of the Orange River in 1867. Soon afterwards the discovery of gold in Pilgrim's Rest and Barberton followed, leading to the establishment of a promising South African economy. The real watershed, however, was the discovery of the Witwatersrand gold deposits which precipitated the Anglo-Boer War. After the war, mine production became more efficient, leading to increased industrial support and promoting economic growth. Gold production in South Africa peaked in 1970 with a 68 per cent contribution to the world-wide gold production.

Apart from gold, South Africa has become the leading producer of chrome, manganese, platinum, vanadium and vermiculite. Needless to say, mining has had and will most likely continue to have a major impact on the South African economy. In the remainder of this chapter an economic perspective on mining will be provided, and this will be followed by an overview of the impact that mining has had on scientific and technological advances in South Africa. Of specific interest to this study is the use of *resource constrained project scheduling models and algorithms* for improving mine production planning, taking into account economic factors, infrastructure limitations, ore body distribution and resource requirements. The research methodology and contributions related to the use of mathematical optimisation will be presented below, and this will finally be followed by an outline of the remainder of the dissertation.

1.1 An economic perspective

Mining in South Africa currently contributes about 7.7% directly to the *Gross Domestic Product* (GDP) [23]. Accounting for the indirect impact due to mining suppliers or downstream consumers, however, this percentage is more likely to be 17% of South Africa's GDP. The total annual mineral sales for South Africa was recorded as R391.4 billion in 2015, of which 69.3% came from exports. The resulting contribution to the national fiscus in the form of taxes and royalties amounted to R11.3 billion. Mining accounted for approximately 10.8% of total fixed investment in 2015 and the total workforce employed by the mining industry was approximately four hundred and sixty thousand, with total employee earnings of R112 billion. These figures exclude the number of jobs created by the secondary mining industries. According to the Chamber of Mines of South Africa [22], for every mining job that is created, two further jobs are created in other sectors of the economy.

According to the 2015 annual report compiled by the Chamber of Mines of South Africa [23], the mining industry has been under severe strain due to increased employment costs, which have been rising at a greater rate than inflation, and a decline in global commodity prices. A cost breakdown of the total mining industry reflects the challenges faced by the mining companies to remain profitable. Almost 50% of mining costs, based on total industry figures, are due to operational and labour costs. This can most likely be attributed to the complex and dangerous environment of typical South African mines, specifically underground gold mines.

In view of the increasing challenges faced by the mining industry, it has become imperative for mining companies to excel in their planning and operational efforts in order to be more cost-efficient. More specifically, with volatile markets which may influence commodity prices on an almost daily basis, the sustainability of mining depends on the agility with which mine production planning is done. With shorter turn-around times in production planning, mining companies may react quicker to global market events by changing their business strategies accordingly.

1.2 Mine production planning

The initial setup of a typical South African underground mine requires detailed *mine layout planning* based on exploration data, detailing the characteristics of the mineral deposits. Once in operation, continuous planning of the mine layout is required, since further information about the mineral deposits is unlocked during the underground excavation process. From an operational point of view, planning is done to manage the resources required for all mining activities. In its simplest form, operational mine planning entails the *scheduling of mining activities* such that the most economical value is derived while coping with physical limitations due to the properties of the mine infrastructure, as well as resource requirements. Different time horizons are used in planning for different purposes. For instance, short term mine planning is required for detailed activity planning and resource allocation, whereas for longer term planning, production output is profiled for the purpose of budgeting and corporate management.

In summary, the two primary planning activities encountered in mine production planning are mine layout planning and mine activity scheduling. Although the scope of this dissertation is limited to the latter, the mathematical models employed in this dissertation depend on a mine layout plan with accompanying data on the distribution of the mineral deposits.

1.3 Scientific and technological advances

Apart from the economic spin-off from mining, important advances have been made in both science and technology over the years. The well-known geostatistical method called *Kriging* is based on the empirical work conducted by the South African engineer, Danie Krige [55]. The use of Kriging is not limited to mining — its application can be found in all spheres of spatial analysis. Proof of further scientific contributions, specifically within a South African context, can be found by considering the journal publications hosted by the South African Institute of Mining and Metallurgy since 1969.

Planning of underground mine production requires the use of specialised *three-dimensional* (3D) draughting software. Historically, mine layout planning was done by means of engineering drawings, projected onto a two-dimensional plane. With advances in computer technology, however, the use of 3D *computer aided design* (CAD) software became an indispensable tool for engineering design. The development of CAD software customised for mine planning gained momentum during the early 1990s and today's CAD technology encompasses a diverse suite of tools ranging from the modelling of mineral deposits as 3D solids to simulation of underground activities visible as coloured polygons within a 3D space. Enabling the 3D CAD environment are computer algorithms responsible for finding solutions to various mathematical models, *e.g.* rotation of objects in 3D, projecting solids onto two-dimensional planes, finding feasible scheduling solutions, *etc.*

The successful completion of this study was dependent on the use of a 3D CAD system. The software system Mine2-4D [68] was made available for this purpose by MineRP, and was used for generating a mine layout plan and capturing the mineral deposit data.

The use of technology in mine planning has greatly improved the efficiency of mine production all over the world. Apart from improving resource utilisation, the use of 3D CAD systems also promotes flexibility in the mine planning process so that layout plans and schedules can be changed on short notice in order to adapt to changing economic or environmental conditions. This, of course, puts more pressure on the developers of scheduling systems to constantly improve the computational efficiency of their products. Furthermore, as computing abilities improve, the demand for more realistic mathematical models increases. For instance, initial mine scheduling systems were restricted to calculating the starting times of activities based on their durations and sequencing rules that dictate the order in which the activities may be executed. It soon became clear that a solution to such a scheduling problem could render infeasible planning due to capacity restrictions imposed by the physical mine infrastructure. Furthermore, these early scheduling systems were not designed to incorporate financial and economic features, resulting in iterative approaches to improve economic viability of mine planning schedules.

More sophisticated mine scheduling systems exist today that are geared towards generating mining schedules that optimise some pre-specified objective function, *e.g.* the maximisation of *net present value* (NPV), while adhering to mine-specific requirements such as sequencing rules, production capacities and other resource limitations. The primary factor determining the ability of a scheduling system to accommodate mining complexities is the computational efficiency of the *scheduling algorithms* employed. In practice, more sophisticated mathematical models capturing more complex mining environments generally require more computing effort. Therefore, in order for a mine planning system to promote flexibility and agility in the mine planning process, its underlying scheduling algorithms should be efficient in the sense of finding good solutions to complex mathematical models in a reasonable amount of time.

1.4 Research methodology and contributions

The theory of scheduling algorithms dates back to the formalisation of the critical-path scheduling problem by Kelley and Walker [51]. Informally, however, the use of horizontal bar charts, where each bar maps to a construction activity, can be traced back to the early 1900s [99]. Although the advent of practical computer technology during the late 1950s meant that various industrial problems could be solved by computer algorithms, it soon became clear that certain types of problems are inherently difficult, and in some cases even impossible to solve. Even today scheduling problems exist, having no more than 60 tasks to be scheduled, that cannot be solved to optimality within a reasonable time using the latest algorithmic and computer technology [80].

In order to make any contribution with respect to the underground mine scheduling problem, an understanding of the foundation of scheduling theory and the ability to build upon previous successes are required. In the next chapter, an overview of the classification of general scheduling problems will be provided in order to show specifically how the underground mine scheduling problem relates to the well-defined *resource constrained project scheduling problem* (RCSP). Furthermore, it will also be shown that most of the typical underground mining capacity constraints can be accommodated within the RCSP framework. There are, however, problem requirements for which new modelling constructs are required. For example, with the increased use of mechanisation within mining the tracking of resources and the ability to take transfer delays into account require some modifications to existing RCSP models.

The methodology proposed in this dissertation makes use of RCSP models and algorithms to solve the mine scheduling optimisation problem. The following is a summary of the anticipated contributions of this study.

- A thorough literature study is provided to explore i) the latest mathematical advances towards efficient solution approaches in the context of the RCSP, and ii) the technical issues pertaining to underground mine scheduling specifically within a South African context.

- It is shown that the underground mine scheduling optimisation problem is a special case of the well-defined RCSP and that most of the practical issues related to resource management in underground mine scheduling can be accommodated in standard RCSP models. The solution of these models within an exact mathematical framework is motivated by the need to quantify the quality of feasible solutions. Furthermore, an exact framework allows for the implementation of generic side constraints without having to change existing algorithms or implement new ones.

- An empirical research methodology is adopted in this dissertation for the verification of algorithmic contributions. Standard benchmark problem instances found in the literature are used for both computational testing and for deriving new problem instances that have characteristics commonly found in underground mine scheduling problems.

- Three existing mathematical formulations of the RCSP are investigated and their appropriateness for solving the underground mine scheduling optimisation problem are determined. The three formulations are known in the literature as the *time indexed*, the *resource flow-based* and *event-based* RCSP formulations. It is shown through empirical evaluations that the resource flow-based formulation admits solution approaches that are computationally more efficient and hence it is used further as the basis for addressing mining-specific modelling requirements.

- The existing resource flow-based and event-based RCSP formulations are presented in the literature only for the purpose of minimising makespan. New mathematical constructs are proposed in this dissertation in order to extend both formulations so as to accommodate the maximisation of net present value. Computational results are provided to show that the resource flow-based formulation admits more efficient solutions than both the time-indexed and the event-based formulations when considering the maximisation of net present value. This is especially true for randomly generated problem instances that possess mining-specific characteristics.

- Algorithmic improvements, in terms of computing times for the underground mine scheduling optimisation problem, are achieved by modelling considerations in the standard RCSP. A constraint aggregation approach and graph reduction approach are presented which are known to improve computing times significantly due to a reduction in the number of variables and constraints in the RCSP formulation. Additional speed-up is achieved through a newly proposed heuristic which is employed to generate initial feasible solutions.

- A Benders decomposition approach is suggested to improve tractability of large-scale RCSP instances. This is especially useful within the mine scheduling optimisation context where problem instances comprise many activities and where each activity may consume or produce many different resources. Benders feasibility cuts are separated within a branch-and-cut framework and a primal heuristic is proposed for the generation of feasible solutions during processing of the branch-and-bound nodes. Explicit and implicit valid inequalities are proposed and the separation of the implicit valid inequalities is performed as part of the branch-and-cut process.

- Several modelling extensions are proposed to cater for mining-specific requirements. First, the resource flow-based formulation is modified to accommodate resource transfer delays, which is an important modelling requirement when the movement of large machinery and equipment needs to be taken into account. The second extension involves the use of multi-mode activity scheduling, where each mode of an activity is associated with different costs and resource requirements. Multi-mode activity scheduling is adopted as a modelling approach in order to address the practice of selective mining. The final modelling extension addresses uncertainty in the resource requirements of mining activities. The randomness in the resource requirements may be attributed to, for instance, the variability in geology or the inability to excavate underground areas exactly as planned.

- The various modelling extensions mentioned above require algorithmic modifications in order to leverage the advantages offered by the proposed Benders decomposition approach. Changes to the heuristic that produces initial feasible solutions are required as well as changes to the Benders master problem, the separation sub-problem, the valid inequalities and the proposed primal heuristic. Computational results are presented to showcase the efficiency with which the modelling extensions are incorporated into the Benders decomposition framework.

- The final contribution of this dissertation involves a realistic case study based on data from a real underground mine. Several problem instances are derived from the original data set for demonstrating model behaviour and computational efficiency.

1.5 Mathematical preliminaries

The model solution approach adopted in this dissertation is set within an exact mathematical programming framework. It suffices, therefore, to provide an overview of well-known mathematical concepts and terminology in the fields of linear algebra, polyhedral theory, linear and integer programming. This section is not intended as an introduction, but rather as a reference to assist the reader with notation conventions.

Let the sets of real numbers and integers be denoted by $\mathbb{R}$ and $\mathbb{Z}$, respectively. For arbitrary index sets $I = \{1, 2, \ldots, m\}$ and $J = \{1, 2, \ldots, n\}$, the set $\mathbb{R}^n$ or equivalently $\mathbb{R}^{|J|}$, denotes all vectors of dimension n that have components in $\mathbb{R}$, and the set $\mathbb{R}^{m\times n}$ or equivalently $\mathbb{K}^{|I|\times|J|}$, denotes a matrix space of dimension $m \times n$ that has components in $\mathbb{R}$. Let $j \in J$ be an index for the vector $\boldsymbol{x}$ such that $\boldsymbol{x} = (x_j)_{j\in J}$. In the remainder of this dissertation, all vectors are treated as column vectors. The transposed of the vector $\boldsymbol{x} \in \mathbb{R}^n$ is denoted by $\boldsymbol{x}^T \in \mathbb{R}^n$.

A vector $\boldsymbol{x} \in \mathbb{R}^n$ can be expressed as a *linear combination* of the vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k \in \mathbb{R}^n$ if there exist some $\boldsymbol{\lambda} \in \mathbb{R}^k$ such that $\boldsymbol{x} = \sum_{i=1}^{k} \lambda_i x_i$. If, in addition, $\boldsymbol{\lambda} \geq 0$ and $\sum_{i=1}^{k} \lambda_i = 1$, $\boldsymbol{x}$ is called a *convex combination* of the vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k \in \mathbb{R}^n$.

The set $H = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^T\boldsymbol{x} = a_0\}$ with $a_0 \in \mathbb{R}$ denotes a *hyperplane* with gradient $\boldsymbol{a} \in \mathbb{R}^n$ and the set $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^T\boldsymbol{x} \leq a_0\}$ denotes a *halfspace*. The intersection of a finite set of halfspaces defined by the set $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$, with $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, is called a *polyhedron*. An inequality of the form $\boldsymbol{a}^T\boldsymbol{x} \leq a_0$ with $a_0 \in \mathbb{R}$ and $\boldsymbol{a} \in \mathbb{R}^n$ is called a *valid inequality* for a polyhedron P, if $P \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^T\boldsymbol{x} \leq a_0\}$.

A *linear programming* (LP) problem entails finding a vector $\boldsymbol{x}^* \in P = \{\boldsymbol{x} \in \mathbb{R}^n_+ : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ that optimises an objective function of the form $\boldsymbol{c}^T\boldsymbol{x}$. The vector $\boldsymbol{x}^*$ is called a *feasible solution* if $\boldsymbol{x}^* \in P$ and is called an *optimal solution* if $\boldsymbol{c}^T\boldsymbol{x}^* \geq \boldsymbol{c}^T\boldsymbol{x}$ for all $\boldsymbol{x} \in P$ in the case of a *maximisation problem*, or if $\boldsymbol{c}^T\boldsymbol{x}^* \leq \boldsymbol{c}^T\boldsymbol{x}$ for all $\boldsymbol{x} \in P$ in the case of a *minimisation problem*.

The standard form of representing an LP problem is:

$$\begin{aligned} \min\ & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t. } & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

Alternatively, the short notation $\min\{\boldsymbol{c}^T\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \in \mathbb{R}^n_+\}$ is used instead. It should be noted that an LP problem with a minimisation objective can be transformed into an LP problem with a maximization objective (and *vice versa*), an LP problem with unbounded variables can be transformed to an LP problem with non-negative bounded variables, and an LP problem with inequality constraints can be transformed into an LP with equality constraints.

An optimal solution to an LP problem over a polyhedron P will always be at a vertex of P, or a convex combination of some of the vertices of P, provided that P is bounded. Otherwise, the LP problem may be *unbounded* with $\boldsymbol{c}^T\boldsymbol{x} = \pm\infty$, depending on the choice of the objective function $\boldsymbol{c}^T\boldsymbol{x}$.

Associated with each *primal* LP problem

$$\min\{\boldsymbol{c}^T\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{R}^n_+\}$$

is the *dual* problem

$$\max\{\boldsymbol{w}^T\boldsymbol{b} : \boldsymbol{w}^T\boldsymbol{A} \geq \boldsymbol{c},\ \boldsymbol{w} \in \mathbb{R}^m_+\}.$$

The relationship between the primal and dual problems is described by the following well-known theorems:

Theorem 1.1 (Duality theorem)**.** *With regard to the primal and dual problems, exactly one of the following statements is true:*

1. *Both problems have optimal solutions* $\boldsymbol{x}^* \in \mathbb{R}^n_+$ *and* $\boldsymbol{w}^* \in \mathbb{R}^m_+$ *with* $\boldsymbol{c}^T\boldsymbol{x}^* = \boldsymbol{b}^T\boldsymbol{w}^*$*.*

2. *One problem is unbounded, in which case the other must be infeasible.*

3. *Both problems are infeasible.*

Theorem 1.2 (Complementary slackness theorem)**.** *Let* $\boldsymbol{x}^* \in \mathbb{R}^n_+$ *and* $\boldsymbol{w}^* \in \mathbb{R}^m_+$ *be any feasible solutions to the primal and dual problems, respectively. These solutions are optimal if and only if*

$$(c_j - \boldsymbol{a}_j^T\boldsymbol{w}^*)x_j^* = 0 \quad and \quad w_i^*(\boldsymbol{a}^i\boldsymbol{x}^* - b_i)^T = 0 \quad j = 1, 2, \ldots, n,$$

where $\boldsymbol{a}_j$ *denotes the* j*-th column and* $\boldsymbol{a}^i$ *denotes the* i*-th row of* $\boldsymbol{A}$*, respectively.*

The objective of solving an *integer linear programming* (ILP) problem is to find an *integer vector* $\boldsymbol{x}^* \in \mathbb{Z}^n \cap P$ with $P = \{\boldsymbol{x} \in \mathbb{R}^n_+ : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$, that optimises the objective function $\boldsymbol{c}^T\boldsymbol{x}$. The problem obtained by omitting the integrality restrictions of an ILP problem is an LP problem, called the LP relaxation of the ILP problem. The objective of solving a *mixed integer linear programming* (MILP) problem is to find vectors $\boldsymbol{x}^* \in \mathbb{Z}^n \cap P$ and $\boldsymbol{y}^* \in \mathbb{R}^m \cap P$ with $P = \{(\boldsymbol{x}, \boldsymbol{y}) \in {\mathbb{R}_+}^{n\times m} : \boldsymbol{A}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{y} \leq \boldsymbol{b}\}$, so that the objective function $\boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{y}$ is optimised.

Consider a minimisation problem where $\boldsymbol{x}^{\mathrm{U}} \in \mathbb{Z}^n \cap P$ denotes a feasible solution to an ILP problem with associated objective function value $\boldsymbol{z}^{\mathrm{U}} = \boldsymbol{c}^T\boldsymbol{x}^{\mathrm{U}}$, and let $\boldsymbol{x}^{\mathrm{L}} \in P$ be an optimal solution to the LP relaxation with associated objective function value $z^{\mathrm{L}} = \boldsymbol{c}^T\boldsymbol{x}^{\mathrm{L}}$. For a minimisation problem, the quantities z^{U} and z^{L} are referred to as an *upper bound* and a *lower bound*, respectively. For a maximisation problem it is the converse. The quantity $(z^{\mathrm{U}} - z^{\mathrm{L}})/z^{\mathrm{L}}$ is called the *integrality gap*.

1.6 Chapter orientation

In Chapter 2, the RCSP is formally introduced within the context of classical scheduling theory. A literature review of the computational complexity involved in solving the RCSP is presented and references are provided of the main streams of mathematical formulations of the RCSP. A section on recent algorithmic advances is also provided, with the main focus on exact approaches.

An introduction to underground mine scheduling is provided in Chapter 3. In order to gain an understanding of the problem to be solved, an overview of the technical aspects and terminology in underground mining is provided. A small example is used to demonstrate that the underground mine scheduling optimisation problem is a special case of the RCSP. There are, however, some practical considerations in underground mining that do not fit into the standard RCSP framework and have to be addressed by generic mathematical programming formulations.

The existing RCSP formulations in the literature are presented in Chapter 4. New mathematical constructs are provided for the resource flow-based and event-based formulations in order to facilitate the maximisation of net present value. In addition, details are provided of a proposed heuristic that is designed to generate initial starting solutions to the RCSP.

An empirical evaluation of the existing RCSP formulations is presented in Chapter 5. The purpose of this chapter is to provide computational evidence that supports the choice of using the resource flow-based formulation as a basis for solving the underground mine scheduling optimisation problem.

Algorithmic efforts in speeding up computing times of the underground mine scheduling optimisation problem are presented in Chapter 6. The primary contributions of this chapter are the introduction of variable and constraint reduction approaches, as well as a Benders decomposition of the RCSP implemented within a branch-and-cut framework. Computational results are provided to demonstrate the scalability of the the newly proposed approaches when applied to randomly generated problem instances that resemble a typical underground mining environment.

Chapter 7 contains a presentation of various modelling extensions aimed at addressing mining-specific requirements. These requirements include the formulation of resource transfer delays, the modelling of uncertainty in resource consumption or production, as well as the modelling of selective scheduling. Algorithmic modifications are also described that allow for the efficient accommodation of these modelling extensions within the proposed branch-and-cut framework.

The ability of the proposed mathematical models and algorithmic approaches to provide solutions to realistic instances of the underground mine scheduling problem, is demonstrated in a special case study in Chapter 8. The case study involves problem instances derived from a real underground mine scheduling data set.

A final summary and conclusion are presented in Chapter 9. Concluding remarks are provided and suggestions for future work are proposed.

CHAPTER 2

Scheduling algorithms and computational complexity

Contents

Recent advances in computer technology and the progress made in developing new algorithmic ideas, have made it possible to solve some industrial-sized optimisation problems. Certain types of optimisation problems are, however, inherently difficult, and in some cases even impossible to solve to optimality in reasonable time. The basic concepts of computational efficiency are presented briefly in this chapter in an attempt to demonstrate the challenges faced in solving underground mine scheduling problems. Furthermore, the most recent algorithmic advances in solving related scheduling problems are presented in the form of a literature overview.

2.1 The classical scheduling problem

The solution of a classical scheduling problem involves the computation of starting times for a set of *jobs*, which may be processed in one or more *time intervals*. The resulting solution of start times is commonly referred to as a *schedule*. Each job may, however, comprise one or more *operations*, which may be required to be processed by one or more *machines*. To be precise, let $\mathcal{J}$ denote the set of all jobs involved in the current scheduling problem, $\mathcal{O}$ the set of all operations and $\mathcal{M}$ the set of all machines. Assuming that different jobs may comprise different operations, $\mathcal{O}(j) \subseteq \mathcal{O}$ is defined as the set of all operations involved in processing job $j \in \mathcal{J}$. Similarly, let $\mathcal{M}(o)$ be the set of machines required to complete operation $o \in \mathcal{O}$. If, for all operations $o \in \mathcal{O}$ it holds that $|\mathcal{M}(o)| = 1$, then the problem involves *dedicated* machines. Otherwise, if for all operations $o \in \mathcal{O}$ it holds that $\mathcal{M}(o) = \mathcal{M}$, the problem involves *parallel* machines. In the case where the same machine $m \in \mathcal{M}$ is also used in two different operations (not necessarily from the same job), *i.e.* $m \in \mathcal{M}(o_1)$ and $m \in \mathcal{M}(o_2)$, with $o_1 \neq o_2$, then *multi-purpose* machines are involved and each machine is equipped with the appropriate tools to handle different operations.

Scheduling problems involving *multi-processor tasks* entail the simultaneous use of machines in $\mathcal{M}(o)$, with $|\mathcal{M}(o)| > 1$, by an operation $o \in \mathcal{O}$.

Due to the vast number of different scheduling problem types, a classification scheme has been developed over time (Grahaman *et al.* [41]). The classification proceeds according to a three-field parameter list $\alpha|\beta|\gamma$ and is especially useful in discussions related to the computational complexity of different types of scheduling problems, see *e.g.* Brucker [16]. The first parameter in the three-field classification, $\alpha = \alpha_1\alpha_2$, is used to distinguish between scheduling problems having different machine environments. For instance if $\alpha_1 = \circ$, it is an indication that each job must be processed on a dedicated machine, *i.e.* $|\mathcal{M}(o)| = 1$ for a given operation $o \in \mathcal{O}$. Otherwise, if α_1 is assigned a value in the set $\{P, Q, R\}$, it is an indication that parallel machines are involved. More specifically, P represents the use of *identical parallel machines*, Q represents the use of *uniform parallel machines* (*i.e.* processing different operations at different speeds), and R represents the use of *unrelated parallel machines* (*i.e.* processing different operations of different jobs at different speeds). Assigning the values $PMPM$ or $QMPM$ to α_1 distinguishes between multi-purpose machines having identical and uniform speeds, respectively. For $\alpha_1 \in \{\circ, P, Q, R, PMPM, QMPM\}$ it is assumed that each job $j \in \mathcal{J}$ comprises only one operation, *i.e.* $|\mathcal{O}(j)| = 1$. The parameter α_2 is used to specify the number of machines involved. For $\alpha_2 = \circ$ an arbitrary number of machines is assumed.

The second parameter in the parameter list, $\beta = \beta_1\beta_2\beta_3\beta_4\beta_5\beta_6$, is used to classify problems according to the job characteristics. That is, β_1 indicates whether or not preemption (job splitting) is allowed. The parameter β_2 is used to describe the precedence relationship between the different jobs. Furthermore, β_3 is used to specify whether release dates exist for the different jobs and β_4 indicates the required processing times of the different operations. The parameter β_5 specifies any deadlines on the different jobs and β_6 indicates whether or not the problem at hand involves the processing of jobs in batches. For a thorough explanation and accompanying examples on using these different parameters for classification, the reader is referred to Brucker [16].

The third parameter γ relates to the specific objective function used for solving the scheduling problem. For instance, by letting c_j be the completion time of a job $j \in \mathcal{J}$, the objective function $\gamma = \sum C_j = \min \sum_{j \in \mathcal{J}} c_j$ represents the minimisation of the total processing time of the scheduling problem. Or alternatively, the minimisation of the makespan of the schedule is specified by $\gamma = C_{\max} = \min\{\max\{c_j | j \in \mathcal{J}\}\}$.

2.2 Computational complexity

Despite the significant progress made in solving difficult computational problems using advanced computer technology, the solution of certain problems of practical size remains a challenge. The term *computational complexity* may, therefore, loosely be described as meaning the effort required in solving computational problems, measured in computing time. The question is, however, whether the problem is inherently difficult to solve or whether, perhaps, it is being solved using an inefficient algorithm. A theoretical framework, based on the hypothetical Turing Machine [95], is used as a standard to facilitate the analysis of algorithms in general. The framework has allowed researchers over many decades to distinguish mathematically between "easy" and "hard" problems. It should be noted that proving computational complexity is an ongoing endeavor and that there are still many computational problems for which no verdict has been reached.

For the purpose of subsequent discussions it suffices to distinguish between a problem description, a problem instance and a problem formulation. A *problem description* provides details about the problem to be solved. For example, the introductory section of this chapter can be considered as a problem description of the generic scheduling problem. It describes the typical input data, the relationship between the data elements, and the criteria for what constitutes a solution. In fact, the classification scheme outlined above serves as a very compact problem description. A *problem instance*, on the other hand, is the actual data to be used for solving a scheduling problem. It should be noted that for a specific problem description there may be several problem instances. Within the context of underground mine scheduling, the problem description of finding optimal schedules remains the same, even if a scheduling algorithm is applied for different mines, *i.e.* different problem instances. A *problem formulation* is considered to be the mathematical description of the problem. More formally, it is a mathematical model in terms of variables and constraints. In the remainder of this dissertation, the terms problem formulation and model will be used interchangeably.

In order to compare one scheduling algorithm with another in terms of computational efficiency, a hypothetical computer is assumed which performs elementary instructions in unit time. That is, all basic instructions performed by the computer program in the execution of an algorithm are assumed to require the same amount of time. Instead of measuring absolute computing time of an algorithm, a performance measure of an algorithm is obtained by relating the number of elementary instructions used in executing the algorithm, to the size of the problem instance. Consider, for example, the multiplication of 983 by 127 as a computational problem. The "high school" algorithm would require three multiplication steps involving a single digit by a number, followed by three additions involving (2×3)-digit numbers (the effect of shifting). The three multiplication steps require a total of 3^2 instructions and the three addition steps require a total of $2(3^2)$ instructions. That is, $3^2 + 2(3^2)$ instructions. If the multiplication is performed with two 4-digit numbers, then it would require a total of $4^2 + 2(4^2)$ instructions. That is, in general, at most $n^2 + 2n^2$ steps are required to perform the multiplication algorithm. The result is that the "running time" of the multiplication algorithm is now expressed as the function $g(n) = n^2 + 2n^2$, with the number of digits n referred to as the size of the problem instance. It should be noted that the increase in running time with respect to the size of the problem instance is of interest and not the absolute running time of an algorithm. More specifically, the term in $g(n)$ with the highest growth rate is of interest. It suffices, therefore, to express the *worst-case* running time of the multiplication algorithm as $f(n) = n^2$. The relationship between the function $g(n)$ and $f(n)$ is formalised through the *big O notation*.

Definition 2.1. *Let $f(n)$ and $g(n)$ be functions mapping positive integers n onto the positive real numbers. If there exists a constant $c > 0$, such that $f(n) \leq cg(n)$ for a large enough n, then $f(n) = O(g(n))$.*

Using the big O notation, the complexity of the above multiplication algorithm is reported to be $O(n^2)$. It should be noted that other multiplication algorithms exist which exhibit different complexity properties. For instance, a very naive algorithm is to simply add the value 983 to itself 127 times. The complexity of this algorithm is $O(n10^{n-1})$, obviously much worse than that of the high school algorithm. In general, algorithms that have a complexity of at most polynomial order are considered *tractable*, *i.e.* practically solvable. It is customary to refer to these algorithms as *polynomial-time* algorithms since they are *solvable in polynomial time*.

Complexity theory assists in evaluating whether it is worthwhile to seek for a polynomial-time algorithm, or to accept that a problem is inherently hard and that most likely no polynomial-time algorithm exists. The classification of a computation problem according to its complexity requires analysing the *decision version* of the problem. For instance, a decision version of the scheduling problem $P|prec|C_{max}$ is to determine whether a feasible solution exists such that

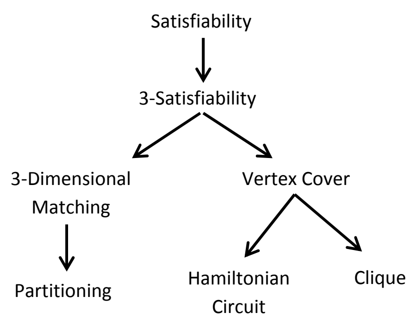


FIGURE 2.1: *An example of reduction of well-known $\mathcal{NP}$-complete problems.*

$C_{max} \leq K$ for some arbitrary K. That is, the answer to a decision problem is either "yes" or "no'. It should be noted that some computational problems are already in a decision format, *e.g.* given the numbers x and y, is y divisible by x?

Definition 2.2. *Let $\mathcal{P}$ denote the class of all decision problems that are solvable in polynomial time.*

Working with the decision version of an optimisation problem, *e.g.* the one described above for the problem $P|prec|C_{max}$, turns out to be just as difficult as solving the original problem. It may, in fact, be shown that, if no polynomial-time algorithm exists for solving the decision version of an optimisation problem, then such a problem cannot be in $\mathcal{P}$. In order to classify these problems, the existence of a problem instance is required for which a "yes" answer is certain. That is, for the scheduling example, a feasible solution for which $C_{max} \leq K$ is a *certificate* that can be tested for validity.

Definition 2.3. *Let $\mathcal{NP}$ denote the class of all decision problems for which their certificates can be tested for validity in polynomial-time.*

It should be clear that $\mathcal{P} \subseteq \mathcal{NP}$ since testing the validity of the certificate of a decision problem, which is in itself polynomial time solvable, would also require at most a polynomial number of steps. There exists another subset of $\mathcal{NP}$ which is called the set of $\mathcal{NP}$-complete problems. To formalise this complexity class, the concept of *polynomial reduction* is introduced.

Definition 2.4. *Let D_1 and D_2 be decision problems with d_1 a problem instance of D_1 that would result in a "yes" answer. Then, D_1 reduces to D_2, if there exists a polynomial-time algorithm g such that $d_2 = g(d_1)$ is a problem instance of D_2 also resulting in a "yes" answer.*

Definition 2.5. *A decision problem $D \subseteq \mathcal{NP}$ is $\mathcal{NP}$-complete if any problem in $\mathcal{NP}$ reduces to D.*

The significance of the $\mathcal{NP}$-complete class is that if a problem $Q \in \mathcal{NP}$-complete can be solved in polynomial time, then all problems in $\mathcal{NP}$ are solvable in polynomial time.

Definition 2.6. *Denote by $\mathcal{NP}$-hard the class of all optimisation problems for which their decision versions are $\mathcal{NP}$-complete.*

In order to prove that a given optimisation problem is $\mathcal{NP}$-hard, it needs to be shown that an existing $\mathcal{NP}$-complete problem reduces, according to Definition 2.4, to the decision version of the given optimisation problem. The ground work for proving the complexity of problems

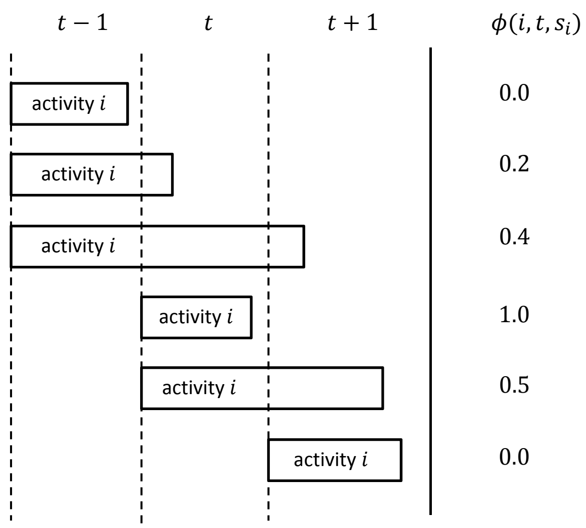


FIGURE 2.2: *Possible scheduling outcomes for an activity i relative to a period t. The function $\phi(i,t,s_i)$ provides the proportion of resources being consumed by activity $i \in \mathcal{N}$ during time period $t \in \mathcal{T}$, given its start time s_i.*

was done by Cook [25], who demonstrated that a well-known combinatorial problem, called *satisfiability* (SAT), is $\mathcal{NP}$-complete. Subsequently, establishing the $\mathcal{NP}$-completeness of many other optimisation problems was achievable by showing that SAT reduces directly, or indirectly, to these problems. Figure 2.1 is an example of the reduction relationship between SAT and other well-known problems proven to be $\mathcal{NP}$-complete.

Examples 3.3 and 3.4 from [16] illustrate the use of reduction to show that the scheduling problems $F3||C_{max}$ and $P2|prec; p_i \in \{1,2\}|C_{max}$ are $\mathcal{NP}$-hard, by showing that the Partitioning problem is reducible to the decision version of $F3||C_{max}$ and the Clique problem is reducible to the decision version of $P2|prec; p_i \in \{1,2\}|C_{max}$.

2.3 The resource constrained project scheduling problem (RCSP)

In the remainder of this dissertation, reference will be made to the scheduling of *activities* rather than the scheduling of jobs in order to be more in-line with current literature on resource constrained scheduling. Let $\mathcal{N}$ denote the index set of all activities and let d_i be the duration of an activity $i \in \mathcal{N}$. The order in which the activities have to be scheduled is specified by means of a directed acyclic graph, called a precedence graph. From this graph the index set $\mathcal{P}(i) \subseteq \mathcal{N}$ may be derived as the set of all predecessor activities of $i \in \mathcal{N}$. The execution of an activity implies that one or more resources will be consumed. For this purpose, $\mathcal{R}$ is defined as the index set of all resources and v_{ir} as the quantity of resource $r \in \mathcal{R}$ being consumed by activity $i \in \mathcal{N}$ over its entire duration. The availability of resources is in most cases restricted and, therefore, U_r is defined as an upper bound on the consumption of resource $r \in \mathcal{R}$.

Finding a solution to the RCSP entails determining starting times for each of the activities such that all precedence rules are obeyed and resource usage per time period is within the specified limits. For that purpose, $s_i \geq 0$ is defined as the starting time of an activity $i \in \mathcal{N}$. It should be noted that, depending on the specific formulation approach (see Section 2.3.1 below), the starting time s_i does not necessarily have to coincide with the start times of the scheduling periods. Furthermore, to facilitate a review of the literature below, involving different ways of formulating resource constraints, the generic notation $\phi(i, t, s_i)$ is used to express the proportion of resources being consumed by activity $i \in \mathcal{N}$ during time period $t \in \mathcal{T}$. Figure 2.2 is a conceptual illustration of how the function $\phi(i, t, s_i)$ provides the proportion of resources being consumed by activity $i \in \mathcal{N}$ during time period $t \in \mathcal{T}$, given the start time s_i of the activity.

A conceptual formulation of the RCSP, when minimising the makespan, is to

$$\text{minimise} \quad \max_{i \in \mathcal{N}} \{s_i + d_i\} \tag{2.1}$$

subject to the constraints

$$\begin{aligned} s_i - s_j &\leq -d_i, && j \in \mathcal{N},\ i \in \mathcal{P}(j), && (2.2)\\ \sum_{i \in \mathcal{N}} \phi(i, t, s_i) v_{ir} &\leq U_r, && r \in \mathcal{R},\ t \in \mathcal{T}, && (2.3)\\ s_i &\geq 0. && && (2.4) \end{aligned}$$

If provided with cash flow values c_i for each activity $i \in \mathcal{N}$ and a discount rate α, the objective function of the RCSP, in the case of maximising *net present value* (NPV), is to

$$\text{maximise} \sum_{i \in \mathcal{N}} e^{-\alpha s_i} c_i. \tag{2.5}$$

It was shown by Garey and Johnson [38] that the RCSP, with a single resource and no precedence constraints, is $\mathcal{NP}$-hard. The intuitive expectation that the RCSP with precedence constraints would also be hard is confirmed by Blazewicz *et al.* [14] and by Lenstra and Rinnooy Kan [60].

The complexity status of the RCSP necessitates the exploration of different algorithmic approaches to either improve lower bounds obtained through the relaxation of the original problem, or improve feasible solutions obtained through the use of heuristic approaches, or both. In the next section various classical problem formulations are discussed, and this is followed by a literature overview of recent algorithmic contributions towards solving the RCSP.

2.3.1 Mathematical formulations

Finding solutions to the RCSP, whether approximate or optimal, requires an explicit mathematical model. The use of *mixed integer linear programming* (MILP) as a modelling approach is well suited for the formulation of the RCSP due to the logical decision-making nature of the problem. It should be noted that several different mathematical formulations may exist to address the same problem. These different formulations may be equivalent in terms of representing the feasible region and the objective function of the RCSP, but they may differ in the number of variables and constraints, as well as the efficiency of the underlying algorithm in finding solutions to these models. In the literature, three main classes of RCSP formulations can be found, namely *time-indexed formulations*, *resource flow-based formulations*, and *event-based formulations*.

Time-indexed formulations are based on the discretisation of time. Binary variables, indexed by both an activity and a time period, are used to indicate at which time period each activity will start. An obvious drawback of this approach is that a fixed time horizon is required which may result in an exponential number of variables. Several alternative time-indexed formulations exist, of which the earliest is by Pritsker *et al.* [79]. Their formulation is probably one of the most simplistic, catering for arbitrary precedence and resource availability constraints. The formulation by Mingozzi *et al.* [69] is obtained by considering only feasible sets of activities per time period. An auxiliary variable is introduced to select from these feasible sets only the one that would result in an optimal solution.

In a resource flow formulation, the resource consumption by activities is modelled as a network flow problem. That is, continuous variables are defined that represent the flow of a resource from one activity to the next [5]. Artigues *et al.* [4] modelled the start time of an activity as a continuous variable, while binary decision variables are required to fix the ordering of the activities. The formulations in both the aforementioned papers were driven by an algorithmic approach rather than by an application requirement. Krüger and Scholl [56] adopted a resource flow formulation to model the transfer of resources among multiple projects, while Quilliot and Toussaint [81] used a resource flow formulation to model delays for when resources are transferred from one activity to another.

Event-based formulations of the RCSP rely on the fact that the end time of an activity, or a set of activities, coincides with the start time of another activity, or set of activities. This point in time is called an *event*. In Artigues *et al.* [3] and Koné *et al.* [54], continuous variables are introduced to keep track of the time of each event and binary variables are used to indicate at which event each activity will start and end, respectively. The total number of events is therefore bounded by the number of activities plus one. An earlier version of the event-based formulation involving more variables is given by Zapata *et al.* [102].

Due to the large number of practical applications of the RCSP, the literature available on the topic is vast. Many variants of the RCSP have been developed over the years as a result of having to solve scheduling problems with application-specific requirements. The survey by Kolisch and Padman [52] is a good source to help navigate through the many variants and extensions to the basic RCSP. Specific attention is afforded to objective functions of the RCSP that involve makespan minimisation and NPV maximisation, and the close relationship between their models and solution approaches. In the survey by Hartmann and Briskorn [43] the distinction between some variations of the classical RCSP is made according to the following criteria:

1. **Activity related attributes**. Scheduling problems are described to either have splittable or non-splittable activities (that is, *preemptive* vs. *non-preemptive* scheduling). The latter implies that once an activity has started, it may not be interrupted until it has been completed. Different forms of preemptive scheduling exist, *e.g.* continuous or preemption at discrete points in time.

   The conceptual RCSP formulations above all assume a constant resource consumption rate v_{ir} per time period of a resource r by an activity i. The alternative is to allow for a *varying resource consumption* rate per period. This can simply be achieved by introducing a time index to the resource consumption parameter, *i.e.* adopting the notation v_{irt} instead of merely v_{ir}, provided that a time-indexed formulation is adopted.

   Setup times are associated with certain types of activities. These are of specific concern in production-related problems where machines need to be configured prior to being able to execute an activity. The most basic case is simply to alter the duration of an activity in order to take setup times into account.

   For the classical RCSP a constant resource efficiency is assumed through the parameter v_{ir}, such that an activity will be completed within the specified duration d_i. In *multi-mode* formulations, each mode of an activity will dictate a different duration and a different resource consumption. Part of the optimisation problem then becomes the selection of an optimal mode for each activity.

2. **Resource-related attributes**. Constraint (2.3) above allows a specific resource $r \in \mathcal{R}$, called a *cumulative resource*, to process more than one activity at a time. A *disjunctive resource* can only process one activity at any given time, and this can be achieved simply by requiring that $v_{ir} = U_r$ for all $i \in \mathcal{N}$.

   Resources can also be classified as *renewable* or *non-renewable*. The resource constraint (2.3), where U_r is the total resource available during any given time period $t \in \mathcal{T}$, is an implementation of a renewable resource. That is, on consumption of the resources at the end of period t, the resource is renewed at the beginning of time period $t + 1$. In the case of a non-renewable resource, the total available resource is cumulatively consumed until depletion.

   The resource constraint, as stated in (2.3) above, implies the same resource availability for each time period. The alternative is to allow *varying resource availability* per time period. This can simply be achieved by introducing a time index to the resource parameter, *i.e.* defining U_{rt} as the total amount of resource r available at a given time t.

3. **Scheduling-related attributes**. The precedence relationship in the classical RCSP does not allow for any lags between the completion of one activity and the start of its successor. *Minimal time lag* constraints can be introduced to force a successor activity to be delayed. This is, of course, another way to cater for machine setup times. *Maximal time lag* constraints may be introduced to prevent the unrealistic delay of activities into the future.

   The earliest time at which an activity is allowed to start may be implemented by a *release date*. That is, even if such an activity's predecessor has been completed, the activity may only start after the release date. Implementing a *deadline* for an activity would force it to be completed before the deadline date, provided its predecessor was completed in time.

   The concept of *time-switching* constraints can be introduced to cater for periods of work and rest within a schedule.

   Constraints can be implemented to enforce *activity-specific policies*. For instance, it may be required that two activities may not start/finish at the same time, or conversely, that the two activities should always start or always finish simultaneously. An extension to this is the modelling of logical operators related to, *e.g.* multiple predecessors. For instance, for a logical "AND" operator, all predecessors to an activity must be completed before the activity itself is allowed to start.

   Constraints related to *resource transferring* may be implemented to capture the delay effects due to the transfer of resources from one location to another. As an extension to this approach, the transfer itself may be represented by another activity, which may in turn consume additional resources.

4. **Objective function alternatives**. The minimisation of the makespan is an example of a *time-based objective* function. Other variations on this theme include the minimisation of lateness or the maximisation of earliness, or possibly even a combination of several time-based measures.

   Objective functions aimed at improving the robustness of a scheduling solution are referred to as *robustness-based objective* functions. One way of deriving a robust schedule is the maximisation of the minimum free slack between the activities. An alternative form is proactive scheduling, which entails the scheduling of backup activities in the event of unplanned delays, *e.g.* machinery replacement in the case of breakdowns.

In contrast to proactive scheduling, *rescheduling objective* functions are used in cases where the project is already under way, but due to unforeseen circumstances, rescheduling of the remaining activities is required with minimal deviation from the original plan.

Renewable and non-renewable resource-based objective functions are driven by (non-) renewable resource-oriented decisions. For example, by associating a unit cost with each resource, a resource investment problem can be formulated that would minimise total (non-)renewable resource cost, provided that a deadline or due date is achieved by the resulting schedule. Resource leveling or smoothing is another important criterion in calculating schedules in an attempt at maintaining a certain level of resource usage.

The presence of cash flows over time, negative or positive, warrants the use of *net present value-based objective* functions. This is of particular interest in scheduling problems for which both costs and revenue generating activities are scheduled. The maximum net present value for a schedule would dictate the acceleration of revenue generating activities and the delay of cost generating activities.

Finally, use of a single type of objective function might be considered impractical, motivating the use of *multiple objective* functions.

2.3.2 Algorithmic solution approaches

Despite the fact that the complexity status of the RCSP has been established as $\mathcal{NP}$-hard, continued efforts in developing new algorithms have facilitated the progressive solution of larger and more difficult problem instances, as evidenced by the success in solving benchmark instances over time. The *project scheduling problem library* [80] was created as a repository of RCSP problem instances and has been referenced extensively over the years to benchmark newly developed models and algorithms. In addition to the problem instances themselves, information is recorded and maintained on optimal solutions or best bounds found thus far. Currently there are four data sets, called J30, J60, J90 and J120, comprising RCSP instances with respectively 30, 60, 90 and 120 activities. Each data set contains 480 different problem instances, except for the J120 data set which contains 600 problem instances. Details on how these problem instances were created can be found in [53]. Although significant progress has been made in solving many of the instances in PSPLIB to optimality, including instances from the J120 data set, at this point in time there are still no optimal solutions for 49 of the J60 problem instances. All of the J30 problem instances have been solved to optimality.

Algorithmic approaches towards solving the RCSP are in general either *exact* or *heuristic*. An exact approach has a finite running time and will terminate with an optimal solution, provided the problem being solved has at least one feasible solution (see Section 1.5 for an overview of optimality conditions and solution bounds). The philosophy adopted in this dissertation is that, although proven optimality may be seen as a luxury in practice, the continuous efforts in devising exact algorithms are important in terms of understanding problem-specific properties that may lead to the development of improved approaches, whether exact or heuristic. Furthermore, from a practical point of view, the solvers used for solving general MILP problems provide solution bounds when prematurely terminated, which may serve as a quality certificate for the incumbent feasible solution. In a mine scheduling optimisation context, the ability to have a quality guarantee on a solution allows decision makers to be confident in corporate decisions involving billions of Rands. Therefore, in this literature overview, the focus will be on work related to exact approaches.

Most of the exact methods reported for the RCSP are based on either the general branch-and-bound method, proposed by Land and Doig [59], for solving binary linear programming problems, or on implicit enumeration methods incorporating customised branch-and-bound schemes. In the latter approach, implicit enumeration involves the exploration of a branch-and-bound tree, with each node in the tree corresponding to a partial scheduling solution. Pruning of the branch-and-bound nodes is performed based on dominance rules, see for example Johnson [49] and Schrage [85]. Various extensions to the basic enumeration scheme have since been proposed, such as using a breadth-first approach (Stinson *et al.* [92]) or a depth-first approach (Christofides *et al.* [24]) for selecting a new node to be explored. Other state of the art customised branch-and-bound schemes include, for example, the approaches by Brucker *et al.* [18], Demeulemeester and Herrolen [31], Laborie [57] and Mingozzi *et al.* [69]. Customised branch-and-bound algorithms for specifically maximising NPV were suggested by, amongst others, Ecmeli and Erenguc [35], Vanhoucke *et al.* [97] and Yang *et al.* [101].

The early popularity of customised branch-and-bound schemes over the more general binary programming branch-and-bound method, was a result of the inefficiency of the latter approach Moodie and Mandeville [72]. Technological advances and the growing maturity of mathematical programming methodologies has since changed the situation. Computational results of Koné *et al.* [54] show improved computing times of the binary programming branch-and-bound method over the MCS exact approach by Laborie [57]. They also show that for problem instances with a short time horizon, the time-indexed formulation of the RCSP outperforms both the continuous-time and event-based formulations. In the case of longer time horizons, either the MCS or event-based formulations appear to be superior, depending on the specific problem instance. A clear advantage over customised branch-and-bound schemes is an ability to solve the RCSP as a binary linear programming problem, while taking other “business-related” side constraints into account. For instance, binary programming formulations have been proposed for maximising NPV while taking constraints on capital expenditure (Doersch and Patterson [32]) and material use (Smith-Daniels and Smith-Daniels [91]) into account.

Recent advances within an exact framework include the use of constraint programming approaches. In Dorndorf *et al.* [34], consistency tests for the disjunctive scheduling problem were derived for the purpose of facilitating constraint propagation, and in Dorndorf *et al.* [33], constraint propagation techniques were used within a branch-and-bound algorithm for the purpose of reducing the search space. Significant progress in solving many of the open PSPLIB instances is due to Horbach [46]. A satisfiability solver was employed for solving the RCSP problem and it managed to solve 80 of the open J60 instances and 44 open J90 instances to optimality. Schutt *et al.* [87] reported that by using a lazy clause generation approach, they were able to close a total of 60 open instances, of which 20 are from the J120 data set. A similar approach was employed by Schutt *et al.* [88], who managed to solve 631 open instances of the PSPLIB, for the RCSP with generalised precedence constraints. A custom branch-and-bound scheme based on constraint propagation was proposed by Leus and Herroelen [61] which makes use of the resource flow formulation of [4].

An approach is generally considered to be a heuristic if it does not terminate with a proven optimal solution. There are, however, approaches that do terminate with upper and lower bounds that give a quantification of the quality of feasible solutions. The *Lagrangian relaxation*-based heuristic by Möhring *et al.* [71] falls within this category. In their approach, the resource constraints for the RCSP with cost-dependent start times are relaxed and the resulting Lagrangian subproblems are solved as minimum cut problems. Solution information from these subproblems are then used by list scheduling algorithms to generate primal solutions. In Ayala *et al.* [9], a Lagrangian relaxation approach is adopted for computing lower bounds of the resource-constrained modulo scheduling problem. Although the formulation by Mingozzi *et al.* [69] may be hampered by an exponential number of feasible sets, their work on the derivation of lower bounding procedures have paved the way for very successful approaches by others, see *e.g.* Baptiste and Demassey [10] and Brucker and Knust [17]. By *stronger formulation* of the problem, or the application of *valid inequalities* to the LP relaxation, improved lower bound solutions can be obtained. For instance, it was shown by Uetz [96] that the disaggregated version of the precedence constraints by Christofides *et al.* [24] improved the lower bound by 75% for a specific problem instance. Demassey *et al.* [30] used a constraint propagation approach to derive valid inequalities for both sequence-based and time-indexed formulations within a *cutting plane* framework. Instead of only strengthening the initial LP relaxation, a *branch-and-cut* approach can be followed whereby the separation of valid inequalities is performed during the processing of the branch-and-bound nodes, see *e.g.* Padberg and Rinaldi [78]. In Zhu *et al.* [103] a branch-and-cut algorithm is used to solve the multimode RCSP while Sirdey and Kerivin [89] use it to solve a process scheduling problem.

2.4 Summary

The development of the literature on scheduling algorithms has been an ongoing endeavour since the formalisation of the critical-path scheduling problem during the late 1950s (Kelley and Walker [51]). Despite the significant progress made over the years in solving larger scheduling problem instances, the ability to obtain solutions for industrial-sized problems remain a challenge. In this chapter, a very brief overview of the theory of computational complexity was provided, which provides a useful framework to distinguish mathematically between "easy" and "hard" computational problems.

The majority of problems encountered in industry that involve the scheduling of activities, are most likely special cases of the RCSP, since the consumption of resources is typically involved and the order in which activities are scheduled to start is determined by a precedence graph. In this chapter, an overview of the different options in formulating the RCSP was provided as part of a literature study.

The use of heuristics may be considered as an attractive approach for solving the RCSP due to its classification as being $\mathcal{NP}$-hard (Blazewicz *et al.* [14]). The emphasis in this study is, however, on exact solution approaches due to the benefit of being able to add generic constraints to the RCSP formulation and being able to obtain a certificate on solutions calculated as part of the branch-and-bound approach. Furthermore, algorithmic enhancements proposed in later chapters show that it is possible to obtain good solutions to large-sized instances of the RCSP in reasonable time, despite its complexity status.

The application of the RCSP within a underground mining context is discussed in more detail in the next chapter.

CHAPTER 3

The underground mine scheduling optimisation problem

Contents

Excavation of activities in mining are very costly and in order to maintain profitability, optimal use of resources is imperative. This may be achieved first by extracting high-grade minerals through proper mine layout designs, and secondly by the optimal scheduling of mining activities that are expected to maximise *net present value* (NPV), while taking mine-specific constraints into account. These two tasks are typically treated as disjoint from each other and the mine layout is used as input to the scheduling process. Several commercial software packages are available for integrating mine layout design and scheduling within a 3D CAD environment [68].

In this chapter, a brief overview of the technical aspects of underground mine scheduling is provided. A comprehensive discussion on different mining methods and geological modelling is, however, beyond the scope of this dissertation. With a basic understanding of an underground mining operation, various modelling assumptions are presented that facilitate the formulation of mathematical models in later chapters.

3.1 Technical aspects of underground mine scheduling

A concentration of underground mineral deposits is commonly referred to as an *ore body*. Depending on the characteristics of an ore body, the extraction of minerals is performed according to specific mining methods. The primary distinction to be made is *open pit mining* and *underground mining*. In the case of open pit mining, the ore body is usually very localised and close to the surface whereas in underground mining, access to the ore body is gained through underground shafts and tunnels. In typical South African gold and platinum mines, the ore body forms a *reef* that may in some cases be only a few centimetres thick, and depending on its inclination with the surface, may reach depths of up to several kilometres.

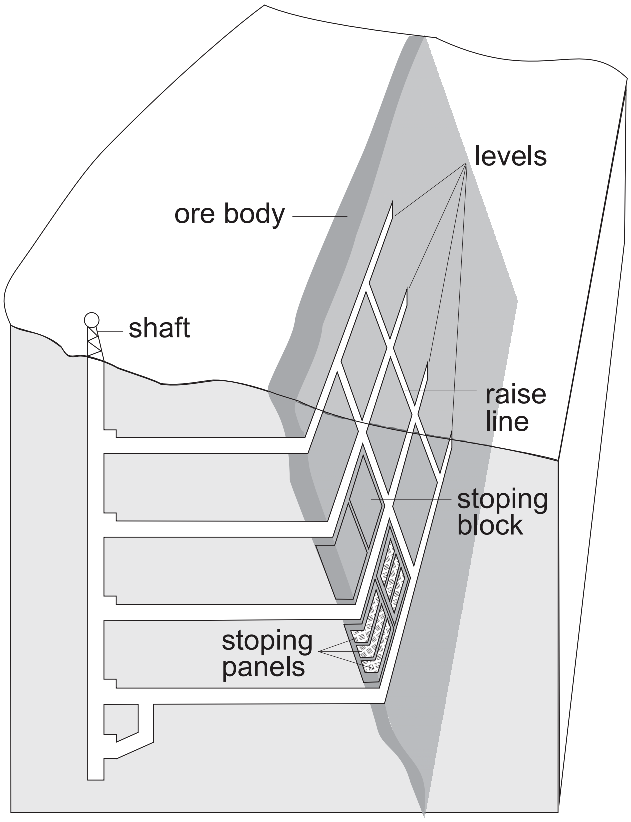


FIGURE 3.1: *A typical underground mine layout.*

The type of underground mining operation to be considered in this dissertation is for a typical South African gold or platinum reef, as depicted in Figure 3.1. Main access to the reef is obtained by a vertical tunnel called a *shaft*. Horizontal tunnels, called *haulages*, are excavated from the shaft in order to provide access to the reef. Haulages running along the plane of the reef at different depths are referred to as *levels* and on each level several smaller tunnels, called *cross-cuts*, provide access to the reef. A *step-over* connects a cross-cut to a tunnel running along the plane of the reef, called a *raise line*. This configuration is illustrated in Figure 3.2 which is a top view where the reef is transparent in the diagram.

Blocks of ore-bearing rock, alongside raise lines and between consecutive levels, are called *stoping blocks* and are demarcated into smaller pieces to form *stoping panels*. The size of the individual stoping panels is determined by the specific mining technique employed as well as by geological constraints. The activity of excavating stoping panels is referred to as *stoping* whereas the excavation of tunnels giving access to the ore body is referred to as *development*. A distinction is made between *off-reef development* and *on-reef development*, where the latter refers to the excavation of tunnels (*e.g.* raise lines) within the ore body, with the result that some minerals are also mined out, but with a high dilution factor. Off-reef development, on the other hand, refers to the excavation of tunnels through waste rock to give access to the ore body.

Different mining methods can be applied for the excavation of stoping panels. Two of the most commonly used mining methods in shallow dip reef mining are *sequential* and *pillar* mining. For a more complete reference on different mining methods, see Hustrulid and Bullock [47]. Figure 3.3 illustrates the differences between sequential and pillar mining. In the latter, parts of the stoping block are left behind as pillars for safety purposes, whereas sequential mining involves clearing out the entire stoping block.

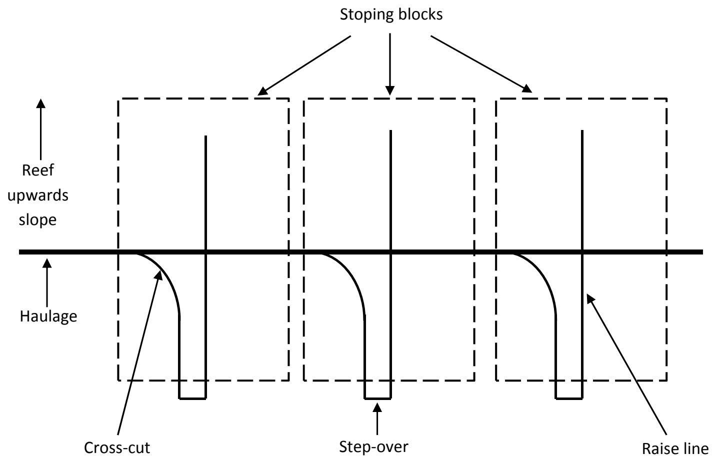


FIGURE 3.2: *Top-view of cross-cuts accessing a reef.*

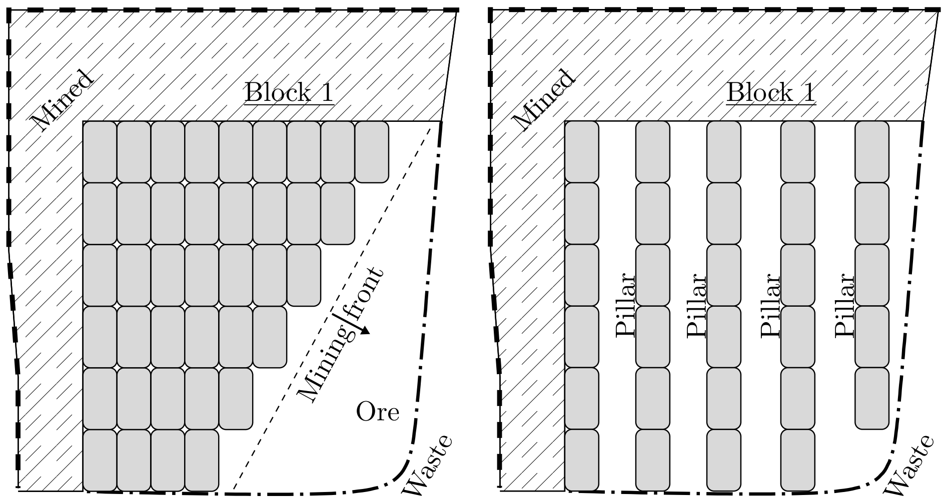


FIGURE 3.3: *Sequential vs. pillar mining (adapted from Kazakidis and Scoble [50]).*

Rock that has been excavated from both development and stoping panels needs to be transported all the way up the shaft. Only the ore from the stoping panels are sent to a processing plant where the valuable minerals are extracted. The routing of rock underground, both waste and ore, is called *tramming*. The different tramming routes are dictated by the mine layout and the location of the shaft.

In reality, the rather ideal layout of a reef, as shown in Figure 3.1, is very unlikely, especially during the initial exploration phases of the ore body. As excavation efforts intensify, more information about the ore body becomes available. The main statistical tool used for estimating the grade distribution of the ore body is called *Kriging* [55]. The output from a Kriging model is a map of the ore body discretised into *evaluation blocks*. A grade estimate is associated with each evaluation block and is calculated through a process of interpolation. The mine planning process commences with an initial mine layout that overlays the evaluation blocks as best possible in order to extract as much as possible of the valuable minerals while leaving behind most of the waste.

The next step in the mine planning process is the discretisation of the mine layout into *mining*

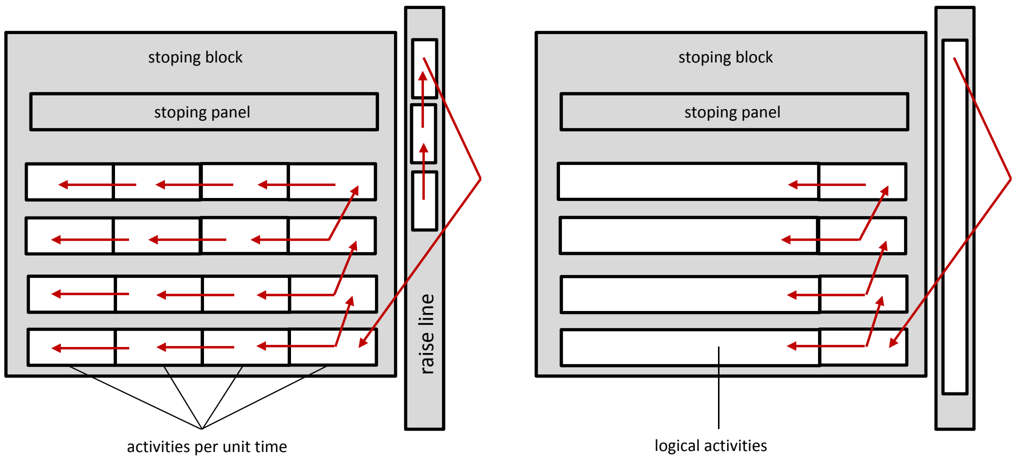


FIGURE 3.4: *Sequencing rules for simulating a sequential mining method. The diagram on the left is for a time-based discretisation and the diagram on the right for a logical discretisation of the activities.*

activities. Traditionally, time-based discretisation was performed to simplify the scheduling of the activities. For instance, if it takes one month to excavate ten metres of an underground tunnel, then the entire tunnel in the three-dimensional design is represented by ten-metre activities, provided a scheduling calendar comprising monthly periods is considered. Alternatively, a functional discretisation entail the creation of logical mining segments that would need to be completed in an uninterrupted fashion. For instance, a logical mining segment may be an entire raise line, or the piece of haulage between two adjacent raise lines, *etc.*

Mining activities are scheduled according to strict precedence rules, commonly referred to as sequencing rules within the mining environment. The physical creation of sequencing rules typically takes place within the same 3D CAD environment in which the mine layout drawing is produced [68]. The application of the sequencing rules is dictated by the mining method employed. For instance, Figure 3.4 shows the placement of sequencing rules to simulate a sequential mining method, for both time-based discretisation and logical discretisation of the activities.

With sequencing rules in place, generating a schedule entails finding starting times for each of the activities such that all precedence rules are obeyed. In order to generate schedules that are feasible with respect to mine-specific constraints and that are optimal with respect to some financially derived objective function, certain properties need to be associated with each of the activities that can be aggregated per scheduling time period. For instance, if the final schedule should be feasible with respect to the capacity of a shaft, specified in terms of tonnes per month, each of the activities should have tonnes associated with it. Depending on the starting time of each activity, the tonnes from each activity are accumulated per period and compared to the shaft capacity to test for feasibility. The final step prior to scheduling is, therefore, the assignment of the necessary properties to each of the activities.

The following physical properties are required in order to solve a capacitated mine scheduling problem:

- **Physical dimension**. This is the physical area (m^2) specified for each activity as part of the mine layout design. In the case of development, the length (m) of the activity is required and in the case of stoping, the stope-width (m) is required.

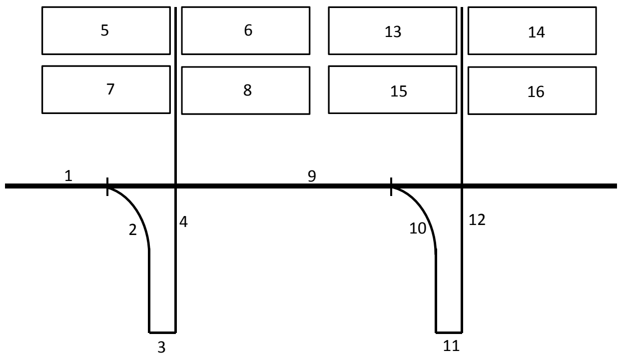


FIGURE 3.5: *A logical discretised section of a mine level. Identification numbers are provided next to each logical mining activity.*

- **Rock density**. This is typically constant for an entire underground mine and is specified as t/m^3, with t measured in metric tonnes.
- **Tonnes**. This measure refers to metric tonnes and is derived from the physical dimension and rock density above.
- **Grade**. This is estimated by the grams of mineral per tonne (g/t).

If the objective being considered is the minimisation of the total project length (*i.e.* the minimisation of the makespan), then the final parameter required for scheduling is the duration of each activity. This depends on the rate at which each of the activities are mined and may depend on several factors, such as the type of the activity (whether development or stoping), the efficiency of the labour resources, the environmental difficulty of excavation, *etc.*

3.2 Solving the mine scheduling optimisation problem as an RCSP

The mathematical formulation of the underground mine scheduling optimisation problem is accommodated by the conceptual formulation of the RCSP (2.1)–(2.4) in Section 2.3. From the discussion above, it is clear that all of the data elements are available to construct the necessary parameter sets for the RCSP formulation. For instance, the set of activities $\mathcal{N}$ is obtained through the discretisation of the mine design layout (either time-based or logical). The set of predecessors $\mathcal{P}(i)$ for an activity $i \in \mathcal{N}$ is derived from the mine sequencing rules. The creation of the resources $\mathcal{R}$ depends on the type of resource constraints considered. For instance, if a shaft capacity has to be imposed in terms of allowable tonnes per month, the resource set $\mathcal{R}$ should include "tonnes" as an element, in which case an upper limit U_r should be defined for this resource. The parameter v_{ir} in the RCSP formulation can now be interpreted as a quantity of resource $r \in \mathcal{R}$ being either consumed or produced. For instance, the scheduling of each activity $i \in \mathcal{N}$ will result in the production of the "tonnes" resource according to the quantity v_{ir}. That is, the value of v_{ir} for an activity i corresponds to the physical tonnes property from the section above, as estimated from the physical dimensions of the activity.

In order to use NPV as an objective function criterion, a cash flow per activity needs to be estimated. Within a mining environment, costs are typically allocated based on activity types.

For instance, the cost of excavating a haulage is expressed in Rand per metre (R/m) and for stoping it is Rand per square metre (R/m^2). A simple way of dealing with activity-based costing within the RCSP is to allocate cost via a resource assignment. For instance, by extending the set of resources $\mathcal{R}$ to include "metres" as an element, a cost parameter c_r is introduced to denote the unit cost of excavating one metre of a haulage. The modified objective function of the RCSP, which includes activity-based costing, is to

$$\text{maximise} \sum_{i \in \mathcal{N}} e^{-\alpha s_i} \sum_{r \in \mathcal{R}} c_r v_{ir}. \tag{3.1}$$

The same mechanism can also be applied to account for the revenue estimates per activity. For instance, the set of resources $\mathcal{R}$ may be extended to include "mineral" as a resource, representing the kilograms (kg) of the specific minerals extracted from the mine. The quantity v_{ir} may be used to indicate how many kilograms of the mineral are produced by activity $i \in \mathcal{N}$ based on the grade, which is one of the physical properties as described above. The mineral price is then captured as the parameter c_r in the objective function (3.1).

To make these concepts more concrete, a small example is considered. Figure 3.5 shows the layout of a section of a mine level consisting of a haulage from which two cross-cuts originate. For each cross-cut, a step-over is followed by a raise line which provides access to four stoping panels.

				Resource quantities (v_{ir})				
i	Type	d_i	$\mathcal{P}(i)$	Off-reef (m)	On-reef (m)	Stoping (m^2)	Tonnes (t)	Mineral (kg)
1	Haulage	200	—	180	0	0	10 000	0
2	Cross-cut	64	1	80	0	0	3 200	0
3	Step-over	10	2	10	0	0	500	0
4	Raise line	90	3	0	60	0	1 800	0
5	Stope panel	225	4	0	0	2 400	6 300	30
6	Stope panel	225	5	0	0	2 400	6 300	30
7	Stope panel	225	6	0	0	2 400	6 300	30
8	Stope panel	225	7	0	0	2 400	6 300	30
9	Haulage	200	1	180	0	0	10 000	0
10	Cross-cut	64	9	80	0	0	3 200	0
11	Step-over	10	10	10	0	0	500	0
12	Raise line	90	11	0	60	0	1 800	0
13	Stope panel	225	12	0	0	2 400	6 300	45
14	Stope panel	225	13	0	0	2 400	6 300	45
15	Stope panel	225	14	0	0	2 400	6 300	45
16	Stope panel	225	15	0	0	2 400	6 300	45

TABLE 3.1: *Activity input values for a small hypothetical example.*

By following a logical discretisation approach of the layout depicted in Figure 3.5, a total of 16 activities are obtained. A capacity restriction of 50 tonnes per day is assumed for this small section of the mine layout and the objective is to find a schedule that maximises the NPV. For this purpose it is assumed that the cost of doing off-reef development is R7 500 per metre, the cost of on-reef development is R5 000 per metre, the cost of stoping is R3 000 per square metre and the current mineral price is assumed to be R550 000 per kilogram. From this information it is clear that several resources will have to be defined, while the production values v_{ir} will need to be specified for each activity-resource combination. Table 3.1 lists the sixteen activities and the first four columns of the table correspond to the activity identification i, the type of the activity, the duration d_i and the predecessor $\mathcal{P}(i)$ of the activity, respectively. For this example only a single predecessor is assumed in each case, whereas in reality there could be an arbitrary number of predecessors. The remaining columns correspond to resources and each cell entry in the table corresponds to the production v_{ir} of a resource $r \in \mathcal{R}$ by an activity $i \in \mathcal{N}$. It is important to note that not all activities produce the same resources. For example, the first row in the table corresponds to the first haulage segment in Figure 3.5. Since this haulage segment is classified as an off-reef development type, there is an entry for it under the resource "Off-reef (m)", but no entries (zeroes) under the resources "On-reef (m)", "Stoping (m^2)" and "Mineral (kg)". The only exception is for the resource "Tonnes (t)" since a capacity is specified for it and the tonnes produced by all of the activities have to be aggregated and tested for feasibility. The only activities contributing to the production of the "Mineral (kg)" resource in this example, are the stoping panels.

In order to capture the production limits on each of the resources, as well as the financial parameters necessary for an NPV calculation, resource data are introduced in Table 3.2. The first column lists the resources $r \in \mathcal{R}$ as they appear in Table 3.1 as the column headings. The unit costs or revenues per resource c_r are provided in the second column. The negative entries denote costs whereas the positive entries denote revenue. The entries in the last column U_r provide the upper limits for each of the resources to be used in the resource capacity constraints.

r	c_r	U_r
Off-reef (m)	−7 500	∞
On-reef (m)	−5 000	∞
Stoping (m^2)	−3 000	∞
Tonnes (t)	0	50
Mineral (kg)	550 000	∞

TABLE 3.2: *Resource input values for the small hypothetical example described in Table 3.1.*

The underground mine scheduling optimisation problem, as captured by the information in Tables 3.1 and 3.2, can now be formulated conceptually as

$$\text{maximising} \quad \sum_{i \in \mathcal{N}} e^{-\alpha s_i} \sum_{r \in \mathcal{R}} c_r v_{ir} \tag{3.2}$$

subject to the constraints

$$s_i - s_j \leq -d_i, \quad j \in \mathcal{N},\ i \in \mathcal{P}(j), \tag{3.3}$$

$$\sum_{i \in \mathcal{N}} \phi(i, t, s_i) v_{ir} \leq U_r, \quad r \in \mathcal{R},\ t \in \mathcal{T}, \tag{3.4}$$

$$s_i \geq 0. \tag{3.5}$$

The explicit formulation of the above problem will be given in the next chapter. For now it suffices to know that this problem can be solved to optimality using standard mathematical programming technology. The most important result expected by a mine planner from an automated scheduling approach is the graphical display of the scheduling solution and production profiles. These are used by the mine planner for validating the solution in terms of feasibility. If the mine planner is satisfied that the solution constitutes a feasible schedule, the next consideration would be the value of the solution in terms of NPV. A solution to the above optimisation problem comprises starting times s_i for each activity $i \in \mathcal{N}$. Using these starting times together

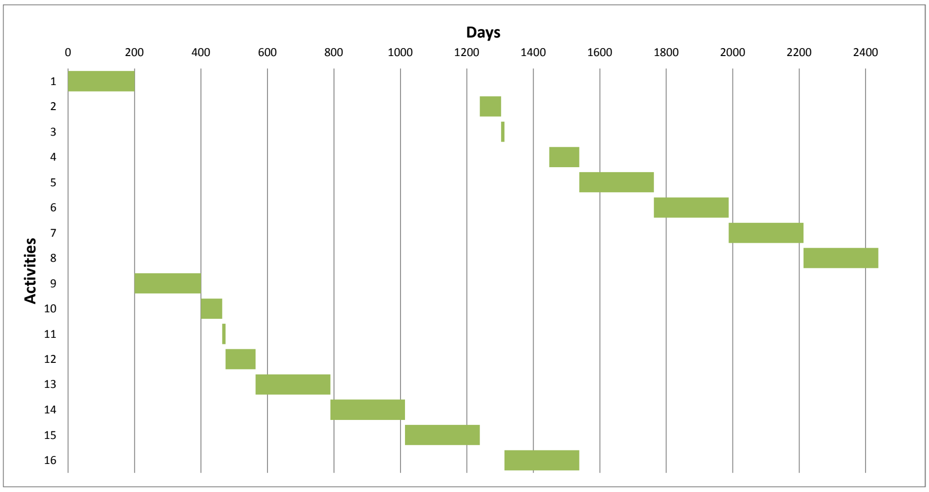


FIGURE 3.6: *Gantt chart showing results for the small hypothetical example considered.*

with the duration of each activity provided as input, a Gantt chart can be created, as illustrated in Figure 3.6.

The graphical nature of a Gantt chart makes it a valuable tool for accessing the feasibility of scheduling solutions. For example, Figure 3.6 shows that all predecessor rules are obeyed, since for this particular section, no parallel activities can occur from a cross-cut (activities 2 and 9) onwards, but the two cross-cuts and their branches may be executed simultaneously. It is also interesting to observe that in order to maximise NPV, it is suggested that the stoping panels that are accessed via cross-cut 9 should be excavated sooner than the stoping panels that are accessed via cross-cut 2. This is in line with the input data which show that there is less mineral content in the latter. That is, the optimisation model prefers to incur revenue as soon as possible and delay costs for as long as possible, which is exactly what one would expect from a function that maximises NPV.

In addition to validating the solution against the given predecessor rules, the aggregated resources produced per time period by all of the activities should be checked against the resource upper limits. Specifically, for the small hypothetical example considered here, the constraint of 50 tonnes per day for the "tonnes" resource, as specified in Table 3.2, may not be exceeded. The graph depicted in Figure 3.7 shows the "tonnes" resource aggregated per day for all of the activities. The graph shows that the capacity constraint of 50 tonnes per day is satisfied by the optimal solution for the small example.

3.3 Practical considerations in mine scheduling optimisation

For the purpose of operational and strategic planning by mine management, a production report is created from a scheduling solution that is expressed in terms of the starting times s_i, for each activity $i \in \mathcal{N}$. Production profiles, such as the one in Figure 3.7, can be generated for different time period sizes, *i.e.* for different *scheduling calendars*. For strategic decision making, reporting typically occurs according to either annual or monthly scheduling calendars, whereas for operational purposes, reporting usually occurs according to a higher time period resolution, such as weekly or even daily calendars. In terms of the conceptual underground mine scheduling model (3.2)–(3.5), the choice of scheduling calendar should only affect reporting and not the solutions that are obtained for the starting times s_i. This is, however, not the case for certain

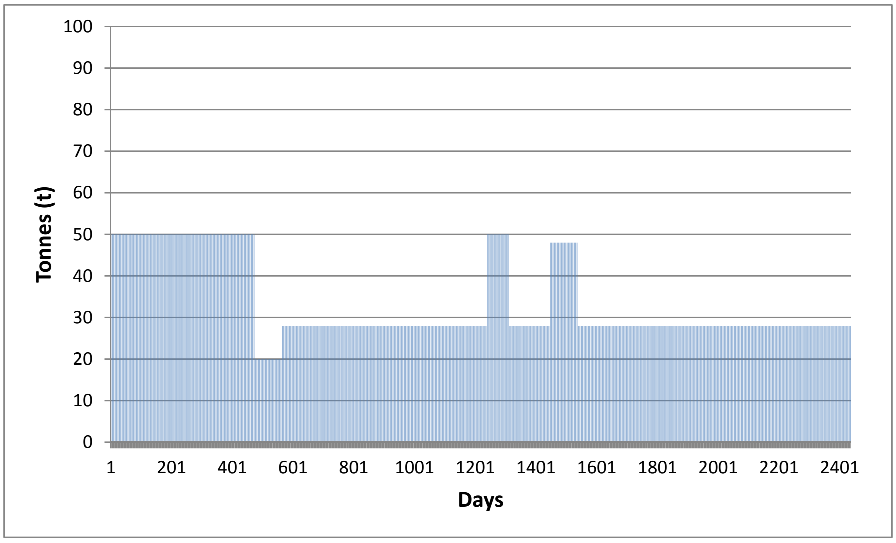


FIGURE 3.7: *Production profile of the "tonnes" resource for the small hypothetical example considered.*

explicit mathematical formulations of the RCSP. If a time-indexed formulation is used (see the literature overview in Section 2.3.1), the starting times s_i are forced to align with the starting times of the calendar time periods.

When using a scheduling calendar with shorter time periods for reporting, it is not required that the discretisation of the mine layout be performed according to a higher resolution. If an activity is scheduled, which overlaps one or more periods, the contribution of the activity to each smaller time period may be calculated proportionately by considering the duration of the activity and its starting time relative to the said time period. There is, however, benefit to a finer grained discretisation, specifically in cases where high variability in the grade distribution is observed. In such cases the optimisation model can pick out parts of a stoping panel with high mineral content first and delay the excavation of the less valuable parts until later. This will, of course, have a positive effect on the NPV calculation. The other benefit of a finer grained discretisation is the ability to interrupt activities. For example, by breaking up a haulage into smaller pieces, its excavation can be delayed midway if insufficient capacity is available and the NPV for that period can be boosted by rather excavating a stoping panel.

In the example presented above, only a single revenue generating resource was considered, namely the "mineral" resource in Table 3.2. In reality, more than one mineral is typically processed at a mine, with each mineral having its own grade and price. This may easily be accommodated within the RCSP framework by simply adding a resource for each mineral. Furthermore, although no mention was made of plant processing costs in the above example, it is possible to accommodate such costs by introducing a resource dedicated to mineral bearing ore sent to the plant. Only mineral-producing activities like stoping and on-reef development will contribute to this resource. In addition to processing costs (R/t), plant capacity ($t/month$) may be specified as the resource upper limit.

Several operational constraints need to be considered in scheduling mine production activities. For instance, there is a tramming capacity per level which limits the volume of rock that can be transported through the haulages. This may simply be achieved by adding a "level" resource such that all activities belonging to this level contribute to the volume of rock for this resource. Since each level will have its own resource, a capacity ($t/month$) may be specified as the resource upper limit.

3.4 Modelling requirements beyond the RCSP framework

It is evident that the RCSP framework is able to incorporate several operational requirements through the use of generic resources. There are, however, some modelling requirements that do not necessarily fit into this framework and which need to be addressed through the use of standard modelling approaches. An incomplete list of some of these requirements are:

- **Fixed vs. variable cost splitting**. The RCSP resource framework, as illustrated by the small example above, facilitates unit costs per resource. That is, only a variable cost component is considered. Fixed costs depend on production volumes and are associated with infrastructure expenses. For instance, if total production in a mine exceeds some threshold, additional ventilation infrastructure is required that will require additional capital injection and will cause ongoing operational expenses to be incurred.

- **Budget constraints**. The planning of production is typically subject to available capital and cash flow requirements. For this reason the allowable operating cost per period would be a very practical constraint to consider.

- **Ore flow networks and blending**. Mining operations that have multiple shafts, multiple processing plants or which operate on multiple reef types, are candidates for scheduling optimisation with a component of ore flow routing. Furthermore, in the context where multiple minerals are being mined from different reef types, a blending problem is encountered where processing plants are configured to accept a certain mix of ore from the different reef types.

- **Equipment and crew logistics**. For every planning horizon considered, a certain level of detail is applicable. The movement of crews or equipment is relevant for short term planning and may have a significant effect on the feasibility of the final schedule. Although work crews and equipment may be viewed as resources with certain upper limits, the key requirement here is to be able to track the movement of crews and equipment, and plan for the delays that will be introduced when moving from one location to another.

- **Selective scheduling**. The standard problem formulations of the RCSP assume that all activities in a given problem instance will be scheduled. Within an underground mining context, selective scheduling may be applied in order to improve profitability by leaving behind low grade mining areas. Precedence constraints are still relevant and by not excavating some parts of a underground mine, access to other parts of the ore body may be limited.

3.5 Related work

Various approaches toward solving the underground mine scheduling problem have been proposed in the literature. The use of MILP approaches have been found by many to be well suited for formulating the mine scheduling optimisation problem. Compared to open pit applications [7, 15, 19, 100], the work done on underground mine scheduling problems is limited. Earlier references to the use of MILP formulations for solving underground mine scheduling problems can be found in Carlyle and Eaves [21], Rahal *et al.* [82] and Smith *et al.* [90], although no model formulations were provided in these papers. MILP formulations for underground mine scheduling problems are presented in Rubio and Diering [83] and Schulze and Zimmermann [86], but without any algorithmic contributions toward improving computing times.

In the paper by Sarin and West-Hansen [84], a MILP formulation is proposed for solving a long-term production scheduling problem in which the allocation of mining units to mine sections are considered. A Benders decomposition approach is followed to improve the solution time of the MILP for small to medium-sized randomly generated problem instances. In Topal [94], two preprocessing algorithms are presented for reducing the number of binary decision variables in the MILP formulation. Also in an attempt to improve computing times, the approach followed by both Little *et al.* [65] and Nehring *et al.* [73] is to aggregate production activities that follow a natural continuous sequence, with the result that the number of variables is reduced. Similarly, the approach followed by Newman and Kuchta [74] is to generate feasible solutions heuristically by aggregating time periods. By doing this, improvements on computing times are achieved, but to the detriment of solution quality. In a follow-up study involving the same mine [67], optimisation-based decomposition heuristics are proposed for solving the corresponding MILP. A heuristic approach is presented by Epstein *et al.* [36] which involves the iterative solution of the linear programming relaxation and the fixing of the binary decision variables.

An improved formulation of the underground mine scheduling optimisation problem is proposed by Terblanche and Bley [93]. The MILP formulation is based on a low resolution time discretisation of the mine layout, while maintaining detailed mineral information through the use of a mining-method-dependent grade tonnage curve. Optimisation-based heuristics are proposed by O'Sullivan and Newman [77] which appear to be successful in generating good approximate solutions within a short time. A good review of underground mine scheduling optimisation can be be found in Newman *et al.* [75].

In the work of Vossen *et al.* [98], a hierarchical Benders decomposition approach is suggested for improving computing times of the open-pit mine block sequencing problem. In their approach two subproblems are formulated in addition to the Benders master problem and a specialised branch-and-bound heuristic is proposed for generating primal solutions. Computational results that were reported show near-optimal solutions could be produced within minutes for medium-sized problem instances.

3.6 Summary

Profitability of mining operations is highly dependent on proper planning and effective resource utilisation. The optimal scheduling of mining activities, with the objective to maximise NPV, plays an important role in achieving this. Before embarking on any kind of optimisation exercise, however, it is imperative to have domain knowledge of mining and to understand the complexities thereof. In this chapter, a brief overview was provided of the technical aspects of underground mine scheduling. Although a comprehensive discussion of different mining methods and geological modelling is beyond the scope of this dissertation, some aspects related to typical underground mine layout and mining operations were described.

A small hypothetical example was used to illustrate how the underground mine scheduling problem may be accommodated within the RCSP framework. An important consideration, however, is the requirement that certain strategic decisions be incorporated into the mine scheduling optimisation problem in the form of generic constraints. In the chapters to follow, it will be shown that the exact framework provided by MILP solvers makes it possible to incorporate these constraints over and above the constraints needed to capture the underlying resource scheduling problem.

CHAPTER 4

Mathematical Models

Contents

Underground mine scheduling optimisation, in its simplest form, entails the scheduling of mining activities to be executed in future in order to maximise NPV, while taking resource and mining-specific constraints into account. It is, therefore, considered to be a special case of the RCSP. In Chapter 2, it was mentioned that three main classes of RCSP formulations exist in the literature, namely *time-indexed formulations*, *resource flow-based formulations*, and *event-based formulations*. Solutions to the underground mine scheduling optimisation problem may thus be computed by using any of these formulations. It should be noted, however, that the time-indexed formulation is the only formulation which naturally accommodates a linear objective function that maximises NPV. Both the resource flow-based and the event-based formulations rely on the maximisation of a non-linear function in order to maximise NPV.

The aim in this chapter is to examine the computational properties of the above-mentioned RCSP formulations. In addition, new mathematical constructs are proposed to extend both the resource flow-based and event-based formulations to be able to accommodate the maximisation of NPV through the use of a linearisation approach. Finally, a preprocessing approach is described which improves the bounds on some of the variables for the three RCSP formulations and which may be used for generating initial feasible solutions.

4.1 General notation

The above-mentioned formulations of the RCSP have several modelling concepts in common. The general notation applicable to all subsequent formulations is presented here. Some of the notation has already been presented in Section 2.3 and will be repeated here for the sake of completeness.

Let $\mathcal{N}$ denote the index set of all activities. An activity $i \in \mathcal{N}$ may, for example, relate to the excavation of part of an underground tunnel, or the placement of machinery that will enable the excavation. The duration of activity $i \in \mathcal{N}$, measured in days, is given by d_i. The execution of an activity implies that one or more resources will be consumed/produced and for this purpose $\mathcal{R}$ is defined as the index set of all resources. A potential resource $r \in \mathcal{R}$ may, for example, include the total on-reef and off-reef tonnes of production, the volume of minerals produced, or the amount of explosives consumed for blasting. Let v_{ir} be a numerical value for the quantity of resource $r \in \mathcal{R}$ being produced/consumed by activity $i \in \mathcal{N}$, over its entire duration.

A non-cyclic precedence graph $H(\mathcal{N}, \mathcal{Z})$, with vertex set $\mathcal{N}$ and arc set $\mathcal{Z}$, is required to express the precedence requirements of the RCSP. Each arc $(i, j) \in \mathcal{Z}$ corresponds to the precedence relation dictating that activity $j \in \mathcal{N}$ should be preceded by activity $i \in \mathcal{N}$.

In order to facilitate the formulation of the various RCSP models below, a source and a sink activity are introduced. If the index set of activities is defined as $\mathcal{N} = \{0, 1, \ldots, N\}$, then the index 0 is used to denote the source activity and the index N is used to denote the sink activity. For all the activities $j \in \mathcal{N} \setminus \{0\}$ that do not have a predecessor, according to the precedence graph $H(\mathcal{N}, \mathcal{Z})$, the arcs $(0, j)$ are added to the set $\mathcal{Z}$. For all the activities $i \in \mathcal{N} \setminus \{N\}$ that do not have a successor, according to the precedence graph $H(\mathcal{N}, \mathcal{Z})$, the arcs (i, N) are added to the set $\mathcal{Z}$.

Let $E_i^{\mathcal{N}}$ be the earliest start time and let $L_i^{\mathcal{N}}$ be the latest start time of an activity $i \in \mathcal{N}$. The earliest and latest start times of an activity are functions of the precedence graph and the planning horizon. The calculation of $E_i^{\mathcal{N}}$ and $L_i^{\mathcal{N}}$ is considered as part of preprocessing.

4.2 Minimisation of makespan

Although the maximisation of NPV is the criterion of choice in underground mine scheduling optimisation, it is convenient to begin the discussion by considering the objective of minimising makespan. This will allow for direct comparison of results obtained in this chapter, with existing findings in the literature. Furthermore, the discussion below serves as an overview since no new mathematical constructs are required to formulate the minimising of makespan for the three RCSP formulations.

4.2.1 The time-indexed RCSP (TI-MIN)

The formulation of the time-indexed RCSP, when considering the minimisation of makespan, is based on the model by Pritsker *et al.* [79]. Let $\mathcal{T} = \{1, 2, \ldots, |\mathcal{T}|\}$ denote the time period indices. The number of periods $|\mathcal{T}|$ depends on the choice of the scheduling calendar. More specifically, if a *daily* scheduling calendar is used it will have more time periods compared to a *monthly* scheduling calendar, provided the same planning horizon is considered. The standard calendar options within the mining environment are annual, quarterly, monthly, weekly and daily calendars. Let $S_t^{\mathcal{T}}$ be the start time of period $t \in \mathcal{T}$. The planning horizon is measured

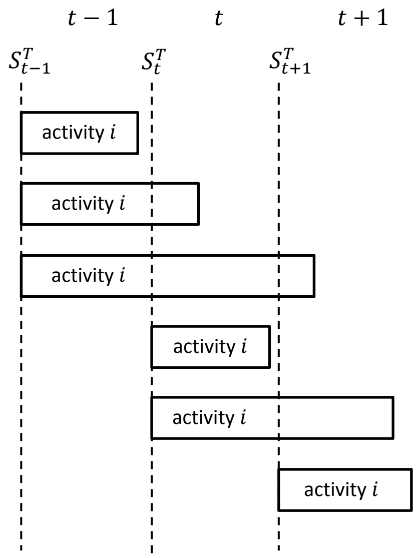


Figure 4.1: *Possible scheduling outcomes for an activity i relative to a period t.*

in terms of days and the start time of a period is calculated as the number of days relative to time zero. The start and end times of periods are dictated by the choice of scheduling calendar and are necessary to accommodate the composition of mixed calendars, *e.g.* first monthly, then quarterly and finally annually. The motivation for using a mixed calendar stems from the uncertainty associated with underground resources that are accessible only far into the future. It does not make sense to schedule according to a daily calendar in ten years' time when there is so much uncertainty in measuring available resources by then. Let p_t be the duration in days of the t-th time period, calculated as $p_t = S_{t+1}^{\mathcal{T}} - S_t^{\mathcal{T}}$. The set $\mathcal{T}(i) \subseteq \mathcal{T}$ denotes the set of eligible start time periods for activity $i \in \mathcal{N}$ based on the earliest and latest start times $E_i^{\mathcal{N}}$ and $L_i^{\mathcal{N}}$, respectively. Specifically, for each activity $i \in \mathcal{N}$ the inequality $E_i^{\mathcal{N}} \leq S_t^{\mathcal{T}} \leq L_i^{\mathcal{N}}$ must hold for all $t \in \mathcal{T}(i)$. Let $\mathcal{N}(t) \subseteq \mathcal{N}$ be the set of activities eligible to start at time period $t \in \mathcal{T}$ based on the earliest and latest start times. That is, $E_i^{\mathcal{N}} \leq S_t^{\mathcal{T}} \leq L_i^{\mathcal{N}}$, for all $i \in \mathcal{N}(t)$.

The variable $x_{it} \in \{0, 1\}$ is used to indicate the start time of an activity. That is, $x_{it} = 1$ indicates that activity $i \in \mathcal{N}$ is scheduled to start at the beginning of time period $t \in \mathcal{T}$. The start time of an activity is therefore aligned with the start time of a calendar period. More specifically, $s_i = \sum_{t \in \mathcal{T}} S_t^{\mathcal{T}} x_{it}$, with $s_i \geq 0$ the start time as defined in Section 2.3.

The duration d_i of an activity $i \in \mathcal{I}$ is assumed to be continuous and, within a mining context, may extend over several time periods. Furthermore, in contrast to the time-indexed formulations in the literature [54], d_i is typically not divisible by multiples of the time period durations. Therefore, the formulation of the objective and the resource constraints below are treated differently. The possible ways in which an activity may extend over time periods are illustrated in Figure 4.1, depending on its starting time and its duration. From the top of Figure 4.1, the cases that are of interest are the ones that overlap time period t. Consider the second case where activity i is scheduled to start at time $t-1$, that is $s_i = S_{t-1}^{\mathcal{T}}$, such that its completion extends into period t. In order to determine the feasibility of a resource constraint for period t, it is necessary to calculate the proportion of resource contribution by activity i to period t. For the second case in Figure 4.1 this proportion is calculated as $(s_i + d_i - S_t^{\mathcal{T}})/d_i$. Note, however, that the fourth case may also be accommodated by using the same formula. Similarly, the third and fifth cases are catered for by means of the formula $(S^{\mathcal{T}}_{t+1} - S^{\mathcal{T}}_t)/d_i$.

The function,

$$\phi(i,t,k) = \begin{cases} \frac{S^{\mathcal{T}}_k + d_i - S^{\mathcal{T}}_t}{d_i}, & \text{if } S^{\mathcal{T}}_k \leq S^{\mathcal{T}}_t \leq S^{\mathcal{T}}_k + d_i, \\ \frac{S^{\mathcal{T}}_{t+1} - S^{\mathcal{T}}_t}{d_i}, & \text{if } S^{\mathcal{T}}_k \leq S^{\mathcal{T}}_t \text{ and } S^{\mathcal{T}}_k + d_i \geq S^{\mathcal{T}}_{t+1}, \\ 0, & \text{otherwise,} \end{cases}$$

which was conceptually introduced in Section 2.3, therefore captures the various cases depicted in Figure 4.1. It returns a resource multiplier which represents the proportion of the resource consumed/produced by an activity i during time period t, based on the duration of the activity and its start period k. Note that the start time s_i of activity i has been substituted with $S^{\mathcal{T}}_k$ in the expression $(s_i + d_i - S^{\mathcal{T}}_t)/d_i$ for the above function.

The objective of the *time-indexed RCSP when minimising makespan* (TI-MIN) is to

$$\text{minimise} \sum_{t \in \mathcal{T}} S^{\mathcal{T}}_t x_{Nt}, \tag{4.1}$$

subject to the constraints

$$\sum_{t \in \mathcal{T}(i)} x_{it} = 1, \qquad i \in \mathcal{N}, \tag{4.2}$$

$$\sum_{t \in \mathcal{T}(j)} S^{\mathcal{T}}_t x_{jt} - \sum_{t \in \mathcal{T}(i)} S^{\mathcal{T}}_t x_{it} \geq d_i, \qquad (i,j) \in \mathcal{Z}, \tag{4.3}$$

$$\sum_{i \in \mathcal{N}(t)} \sum_{\substack{k \in \mathcal{T} \\ k \leq t}} \phi(i,t,k) v_{ir} x_{ik} \leq p_t U_r, \qquad r \in \mathcal{R},\ t \in \mathcal{T}. \tag{4.4}$$

The objective function (4.1) minimises the start time of the sink activity N and the constraint set (4.2) forces each activity to be scheduled in one of the time periods $t \in \mathcal{T}$. Constraint set (4.3) enforces the precedence requirements according to the precedence graph $H(\mathcal{N}, \mathcal{Z})$ and constraint set (4.4) enforces an upper limit on the consumption/production of resources.

4.2.2 The resource flow RCSP (RF-MIN)

The earliest MILP formulation of the flow-based RCSP is due to Artigues *et al.* [4], in which a polynomial insertion algorithm is proposed for solving the RCSP. This formulation is, however, driven by an algorithmic approach and is not formulated for the purpose of solving it by means of an MILP solver. The work by Koné *et al.* [54] is the first to provide numerical results for a resource flow RCSP formulation solved using an off-the-shelf commercial MILP solver.

In order to represent the flow of resources, the graph $G(\mathcal{N}, \mathcal{A})$ is introduced, with the set of arcs $\mathcal{A}$ representing the flow of resources among the nodes in $\mathcal{N}$. Let $\mathcal{A}(i) \subset \mathcal{A}$ be the set of arcs adjacent to the node $i \in \mathcal{N}$. The notation $(i,j) \in \mathcal{A}(i)$ denotes an arc for which node i is the source and the notation $(i,j) \in \mathcal{A}(j)$ denotes an arc for which node j is the target. In order to model the flow of resources, the daily resource requirements of the source and sink activities are set equal to the availability of the resources, *i.e.* , $v_{0r}/d_0 = v_{Nr}/d_N = U_r$, for all $r \in \mathcal{R}$.

The primary decision variables are the starting times $E^{\mathcal{N}}_i \leq s_i \leq L^{\mathcal{N}}_i$ for each of the activities $i \in \mathcal{N}$. In order to formulate the resource transfer constraints, the resource flow variables $f_{ijr} \geq 0$ are introduced to denote the flow of a resource $r \in \mathcal{R}$ from activity $i \in \mathcal{N}$ to $j \in \mathcal{N}$.

The variables $z_{ij} \in \{0,1\}$, called the *linear ordering variables*, are used to indicate the ordering of activities based on the flow of resources. That is, if $z_{ij} = 1$, then activity j must start after completion of activity i, since resources will then be allowed to be transferred from activity i to j.

The objective of the *flow-based RCSP when minimising makespan* (RF-MIN) is to

$$\text{minimise} \quad s_N \tag{4.5}$$

subject to the constraints

$$\begin{aligned}
z_{ij} &= 1, & (i,j) \in \mathcal{Z}, \quad & (4.6)\\
s_j - s_i - (d_i + M)z_{ij} &\geq -M, & (i,j) \in \mathcal{A}, \quad & (4.7)\\
\sum_{(i,j)\in\mathcal{A}(i)} f_{ijr} &= v_{ir}/d_i, & i \in \mathcal{N},\ r \in \mathcal{R}, \quad & (4.8)\\
\sum_{(i,j)\in\mathcal{A}(j)} f_{ijr} &= v_{jr}/d_j, & j \in \mathcal{N},\ r \in \mathcal{R}, \quad & (4.9)\\
f_{ijr} - \min\{v_{ir}/d_i, v_{jr}/d_j\}z_{ij} &\leq 0, & (i,j) \in \mathcal{A},\ r \in \mathcal{R}. \quad & (4.10)
\end{aligned}$$

The objective function (4.5) minimises the makespan by minimising the starting time of the sink activity while constraint set (4.6) is required to ensure feasibility in terms of activity precedence. Constraint set (4.7) is collectively called the *linear ordering constraints* which determine the linear ordering variables z_{ij} based on the starting time s_j of activity j and the completion time of its predecessor i, given by $s_i + d_i$. A reasonable choice for the large number M in (4.7) would be the latest possible finishing time of the schedule according to the calendar, *i.e.* $M = \mathcal{S}^{\mathcal{T}}_{|\mathcal{T}|}$.

The resource flow requirements are imposed by constraint sets (4.8) and (4.9), stating that all the flow of resources into an activity (4.9) and all the flow of resources out of an activity (4.8) should match the daily resource requirement v_{ri}/d_i by an activity i. According to constraint set (4.10), the flow of resources from activity i to j is permitted only if activity j is scheduled to start after the completion of activity i, that is when $z_{ij} = 1$.

According to Artigues *et al.* [3] and Koné *et al.* [54], the following redundant constraints may be added to the above formulation:

$$\begin{aligned}
z_{ij} + z_{ji} &\leq 1, & (i,j) \in \mathcal{A},\ i < j, \quad & (4.11)\\
z_{ij} + z_{jk} - z_{ik} &\leq 1, & (i,j),(j,k),(i,k) \in \mathcal{A},\ i \neq j,\ i \neq k,\ j \neq k. \quad & (4.12)
\end{aligned}$$

Although constraint sets (4.11) and (4.12) are present in both formulations provided by Artigues *et al.* [3] and Koné *et al.* [54], no formal proofs or evidence of improvement in computing times are provided as a result of inclusion of these constraints in the model. For the sake of completeness, it is shown below that these constraints are indeed redundant. Furthermore, computational results presented in the next chapter show that constraint sets (4.11) and (4.12) may, in fact, have a detrimental effect on computing times when included in the formulation of resource flow-related models.

Proposition 4.1. *Constraint set* (4.11) *is redundant, since the same effect is obtained by the linear ordering constraints* (4.7).

Proof. Consider activities i and j with $d_i > 0$ the duration of activity i. If $z_{ij} = 1$, then according to the linear ordering constraints (4.7), $s_i < s_j$. Assume now that $z_{ji} = 1$, then $s_j < s_i$ which is a contradiction, implying that z_{ji} has to be zero. □

Proposition 4.2. *Constraint set* (4.12) *is redundant, since the same effect is obtained by the linear ordering constraints* (4.7)*.*

Proof. Consider activities i, j and k with respective durations $d_i > 0$, $d_j > 0$ and $d_k > 0$. If $z_{ij} = 1$ and $z_{jk} = 1$, then according to the linear ordering constraints (4.7), $s_i < s_j$ and $s_j < s_k$. Assume now that $z_{ki} = 1$, then $s_k < s_i$ which is a contradiction, implying that z_{ki} has to be zero. □

4.2.3 The event-based RCSP (EB-MIN)

The formulation of the event-based RCSP presented here is the "On/Off event-based formulation" suggested by Koné *et al.* [54]. The notion of an "event" is introduced to signify the start of one or more activities. For this purpose the variable $z_{ie} \in \{0, 1\}$ is used to indicate whether the start time of an activity $i \in \mathcal{N}$ coincides with the time of an event $e \in \mathcal{E}$, where $\mathcal{E} = \{0, 1, 2, \ldots, N\}$ denotes the index set of all possible events. The number of events equals the number of activities, indicating that for a worst case scenario each activity will start at a different event. The decision variable $t_e \geq 0$ denotes the time of an event $e \in \mathcal{E}$.

The objective of the *event-based RCSP when considering minimisation of the makespan* (EB-MIN) is to

$$\text{minimise } s_N, \tag{4.13}$$

subject to the constraints

$$\begin{aligned}
s_N - t_e - (z_{ie} - z_{i,e-1})d_i &\geq 0, && e \in \mathcal{E},\ i \in \mathcal{N}, && (4.14)\\
\sum_{e \in \mathcal{E}} z_{ie} &\geq 1, && i \in \mathcal{N}, && (4.15)\\
t_0 &= 0, && && (4.16)\\
t_{e+1} &\geq t_e, && e \in \mathcal{E} \setminus \{N\}, && (4.17)\\
t_f - t_e - (z_{ie} - z_{i,e-1})d_i + (z_{if} - z_{i,f-1})d_i &\geq -d_i, && i \in \mathcal{N},\ e \in \mathcal{E},\ f \in \mathcal{E},\ e \leq f && (4.18)\\
\sum_{e'=0}^{e-1} z_{ie'} - e(1 - (z_{ie} - z_{i,e-1})) &\leq 0, && e \in \mathcal{E} \setminus \{0\}, && (4.19)\\
\sum_{e'=e}^{N-1} z_{ie'} - (N - e(1 + (z_{ie} - z_{i,e-1})) &\leq 0, && e \in \mathcal{E} \setminus \{0\}, && (4.20)\\
z_{ie} + \sum_{e'=0}^{e} z_{je'} - (1 - z_{ie})e &\leq 1, && e \in \mathcal{E},\ (i,j) \in \mathcal{Z}, && (4.21)\\
\sum_{i \in \mathcal{N}} v_{ir}/d_i z_{ie} &\leq U_r, && e \in \mathcal{E},\ r \in \mathcal{R}, && (4.22)\\
E_i^{\mathcal{N}} z_{ie} - t_e &\leq 0, && e \in \mathcal{E},\ i \in \mathcal{N}, && (4.23)\\
t_e - L_i^{\mathcal{N}}(z_{ie} - z_{i,e-1}) - L_N^{\mathcal{N}}(1 - (z_{ie} - z_{i,e-1})) &\leq 0, && e \in \mathcal{E},\ i \in \mathcal{N}. && (4.24)
\end{aligned}$$

The objective (4.13) minimises the start time of the sink node and effectively, minimises the makespan of the project. Note that s_N is the only start time variable defined for the event-based formulation. The constraint set (4.14) is responsible for linking s_N with the event dates and constraint set (4.15) ensures that the start of each activity is aligned with only one event. Constraint sets (4.16) and (4.17) are responsible for event sequencing and constraint set (4.18) links the binary variables z_{ie} with the start times of the events t_e. Constraint sets (4.19) and (4.20) are required to ensure that event start times are contiguous, *i.e.* if an activity $i \in \mathcal{N}$ starts at event $e \in \mathcal{E}$, then activity i cannot be processed before event e. Constraint set (4.21) is necessary to ensure that event start times satisfy the activity precedence requirements and constraints (4.22) are the resource capacity constraints defined for each resource $r \in \mathcal{R}$. Constraints (4.23) and (4.24) ensure that activity start times are within the earliest and latest start time bounds defined by $E_i^{\mathcal{N}}$ and $L_i^{\mathcal{N}}$, respectively.

4.3 Maximisation of NPV

In contrast to the time-indexed formulation which lends itself naturally to be formulated as a linear objective function, the resource flow-based and event-based formulations require optimisation of a non-linear function in order to maximise NPV (see equation (3.1) in Section 3.2). A new mathematical construct is proposed for both the flow-based and event-based formulations in order to allow for the maximisation of NPV.

In the first section below, the time-indexed formulation for the maximisation of NPV is provided, and this is followed by the adapted versions of the flow-based and event-based formulations when considering the maximisation of NPV.

4.3.1 The time-indexed RCSP (TI-MAX)

Keeping with the above notation, the variable $x_{it} \in \{0, 1\}$ is used to indicate the start time of an activity and the function $\phi(i, t, k)$ provides the proportion of resources consumed by an activity i during time period t, based on the duration of the activity and its start period k. The function $\phi(i, t, k)$ is now also required in the objective function in order to calculate the NPV correctly, based on the start time of each of the activities.

The objective of the *time-indexed RCSP when maximising NPV* (TI-MAX) is to

$$\text{maximise} \sum_{i\in\mathcal{N}} \sum_{r\in\mathcal{R}} \sum_{k\in\mathcal{T}} \left(\sum_{t=k}^{S_t^{\mathcal{T}} < S_k^{\mathcal{T}} + d_i} \frac{\phi(i,t,k) c_r v_{ir}}{(1+\alpha)^{S_t^{\mathcal{T}}}} \right) x_{ik}, \tag{4.25}$$

subject to the constraints

$$\sum_{t\in\mathcal{T}(i)} x_{it} = 1, \qquad i \in \mathcal{N}, \tag{4.26}$$

$$\sum_{t\in\mathcal{T}(j)} S_t^{\mathcal{T}} x_{jt} - \sum_{t\in\mathcal{T}(i)} S_t^{\mathcal{T}} x_{it} \geq d_i, \qquad (i,j) \in \mathcal{Z}, \tag{4.27}$$

$$\sum_{i\in\mathcal{N}(t)} \sum_{\substack{k\in\mathcal{T} \\ k\leq t}} \phi(i,t,k) v_{ir} x_{ik} \leq p_t U_r, \qquad r \in \mathcal{R},\ t \in \mathcal{T}. \tag{4.28}$$

The objective function (4.25) maximises NPV at a discount rate of α. The inner summation involving the index t is used to aggregate the NPV values for the time periods over which activity i extends, in the case where activity i is scheduled to start at time period k. The function $\phi(i, t, k)$ is, therefore, used to adjust the revenue or cost for each activity proportionately, based on the start time and duration of the activity.

The constraint sets (4.26)–(4.28) correspond exactly to the constraint sets (4.2)–(4.4) which were described above for the time-indexed formulation used for the minimisation of makespan.

4.3.2 The resource flow RCSP (RF-MAX)

Additional variables and constraints are introduced to facilitate the formulation of an objective function that maximises NPV. Recall that the objective function (3.2), which maximises NPV, is non-linear in the variables s_i and can be written as

$$\text{maximise} \sum_{i \in \mathcal{N}} f_i(s_i), \tag{4.29}$$

where

$$f_i(s_i) = e^{-\alpha s_i} \sum_{r \in \mathcal{R}} c_r v_{ri}. \tag{4.30}$$

A piece-wise approximation of the objective function is suggested to deal with each non-linear function $f_i(s_i)$ according to the approaches by Dantzig [27] and Markowitz and Manne [66]. Let the points (s_{iv}, f_{iv}), $v \in \mathcal{V} = \{0, 1, 2, \ldots, V-1\}$ be the vertices for the piece-wise linear approximation of the function $f_i(s_i)$. The decision variable $y_i \in \mathbb{R}$ is introduced to capture the approximate value of $f_i(s_i)$ according to the piece-wise linear approximation. For that purpose, auxiliary variables $\lambda_{iv} \geq 0$, with $v \in \mathcal{V}$ and $\ell_{iv} \in \{0, 1\}$, are defined for $v \in \mathcal{V} \setminus \{0\}$. The latter serves the purpose of selecting the most appropriate line segment for local approximation with respect to the objective function, whereas the former are needed to express the decision variables s_i and y_i as convex combinations of the knots (s_{iv}, y_{iv}), $v \in \mathcal{V}$.

The objective of the *resource flow-based RCSP when considering maximisation of NPV* (RF-MAX) is to

$$\text{maximise} \sum_{i \in \mathcal{N}} y_i, \tag{4.31}$$

subject to the constraints

$$\begin{aligned}
z_{ij} &= 1, && (i,j) \in \mathcal{Z}, && (4.32)\\
s_j - s_i - (d_i + M) z_{ij} &\geq -M, && (i,j) \in \mathcal{A}, && (4.33)\\
\sum_{(i,j) \in \mathcal{A}(i)} f_{ijr} &= v_{ir}/d_i, && i \in \mathcal{N},\ r \in \mathcal{R}, && (4.34)\\
\sum_{(i,j) \in \mathcal{A}(j)} f_{ijr} &= v_{jr}/d_j, && j \in \mathcal{N},\ r \in \mathcal{R}, && (4.35)\\
f_{ijr} - \min\{v_{ir}/d_i, v_{jr}/d_j\} z_{ij} &\leq 0, && (i,j) \in \mathcal{A},\ r \in \mathcal{R}, && (4.36)\\
s_i - \sum_{v \in \mathcal{V}} \lambda_{iv} s_{iv} &= 0, && i \in \mathcal{N}, && (4.37)\\
y_i - \sum_{v \in \mathcal{V}} \lambda_{iv} y_{iv} &= 0, && i \in \mathcal{N}, && (4.38)\\
\sum_{v \in \mathcal{V}} \lambda_{iv} &= 1, && i \in \mathcal{N}, && (4.39)\\
\lambda_{i0} - \ell_{i1} &\leq 0, && i \in \mathcal{N}, && (4.40)\\
\lambda_{iv} - \ell_{iv} - \ell_{i(v+1)} &\leq 0, && i \in \mathcal{N},\ v \in \mathcal{V} \setminus \{0, V-1\}, && (4.41)\\
\lambda_{i(V-1)} - \ell_{i(V-1)} &\leq 0, && i \in \mathcal{N}. && (4.42)
\end{aligned}$$

The objective function (4.31) expresses the NPV as the sum of the linear piece-wise approximations $y_i \approx f_i(s_i)$, for all activities $i \in \mathcal{N}$. Constraint sets (4.32)–(4.36) are exactly the same as constraint sets (4.6)–(4.10), which where described above for the resource flow problem when minimising makespan. Constraint sets (4.37) and (4.38) express s_i and y_i as convex combinations of the piece-wise linearisation knots of $f_i(s_i)$, for all $i \in \mathcal{N}$. Convexity conditions are maintained by (4.39), while constraint sets (4.40), (4.41) and (4.42) are responsible for enabling the convexity variable λ_{iv} to take on an appropriate value based on the selection of a specific line segment ℓ_{iv}.

4.3.3 The event-based RCSP (EB-MAX)

The same linear piece-wise approximation approach described above for the RF-MAX, is followed here to accommodate the maximisation of NPV when considering an event-based RCSP formulation. Recall that the maximisation of NPV involves the non-linear function $f_i(s_i)$ for each activity $i \in \mathcal{N}$. For this purpose the starting time variables s_i are introduced and linked to the starting times of events by making use of additional constraints. The decision variables $y_i \in \mathbb{R}$ are introduced to capture the approximate value of $f_i(s_i)$, given the starting times s_i for each $i \in \mathcal{N}$. Auxiliary variables $\lambda_{iv} \geq 0$, with $v \in \mathcal{V}$ and $\ell_{iv} \in \{0, 1\}$, with $v \in \mathcal{V} \setminus \{0\}$, are required to express the decision variables s_i and y_i as convex combinations of the piece-wise approximation (s_{iv}, y_{iv}), $v \in \mathcal{V}$.

The objective of the *event-based RCSP when considering maximisation of NPV* (EB-MAX) is to

$$\text{maximise} \sum_{i \in \mathcal{N}} y_i, \tag{4.43}$$

subject to the constraints

$$\begin{aligned}
\sum_{e \in \mathcal{E}} z_{ie} &\geq 1, && i \in \mathcal{N}, && (4.44)\\
t_0 &= 0, && && (4.45)\\
t_{e+1} &\geq t_e, && e \in \mathcal{E} \setminus \{N\}, && (4.46)\\
t_f - t_e - (z_{ie} - z_{i,e-1})d_i + (z_{if} - z_{i,f-1})d_i &\geq -d_i, && i \in \mathcal{N},\ e \in \mathcal{E},\ f \in \mathcal{E},\ e \leq f && (4.47)\\
\sum_{e'=0}^{e-1} z_{ie'} - e(1 - (z_{ie} - z_{i,e-1})) &\leq 0, && e \in \mathcal{E} \setminus \{0\}, && (4.48)\\
\sum_{e'=e}^{N-1} z_{ie'} - (N - e(1 + (z_{ie} - z_{i,e-1})) &\leq 0, && e \in \mathcal{E} \setminus \{0\}, && (4.49)\\
z_{ie} + \sum_{e'=0}^{e} z_{je'} - (1 - z_{ie})e &\leq 1, && e \in \mathcal{E},\ (i,j) \in \mathcal{Z}, && (4.50)\\
\sum_{i \in \mathcal{N}} v_{ir}/d_i z_{ie} &\leq U_r, && e \in \mathcal{E},\ r \in \mathcal{R}, && (4.51)\\
E_i^{\mathcal{N}} z_{ie} - t_e &\leq 0, && e \in \mathcal{E},\ i \in \mathcal{N}, && (4.52)\\
t_e - L_i^{\mathcal{N}}(z_{ie} - z_{i,e-1}) - L_N^{\mathcal{N}}(1 - (z_{ie} - z_{i,e-1})) &\leq 0, && e \in \mathcal{E},\ i \in \mathcal{N}. && (4.53)\\
s_i - (1 - (z_{ie} - z_{i,e-1}))M - t_e &\leq 0, && e \in \mathcal{E} \setminus \{0\},\ i \in \mathcal{N}, && (4.54)\\
s_i + (1 - (z_{ie} - z_{i,e-1}))M - t_e &\geq 0, && e \in \mathcal{E} \setminus \{0\},\ i \in \mathcal{N}, && (4.55)
\end{aligned}$$

$$
\begin{aligned}
s_i - \sum_{v \in \mathcal{V}} \lambda_{iv} s_{iv} &= 0, & i \in \mathcal{N}, \quad (4.56)\\
y_i - \sum_{v \in \mathcal{V}} \lambda_{iv} y_{iv} &= 0, & i \in \mathcal{N}, \quad (4.57)\\
\sum_{v \in \mathcal{V}} \lambda_{iv} &= 1, & i \in \mathcal{N}, \quad (4.58)\\
\lambda_{i0} - \ell_{i1} &\leq 0, & i \in \mathcal{N}, \quad (4.59)\\
\lambda_{iv} - \ell_{iv} - \ell_{i(v+1)} &\leq 0, & i \in \mathcal{N},\ v \in \mathcal{V} \setminus \{0, V-1\}, \quad (4.60)\\
\lambda_{i(V-1)} - \ell_{i(V-1)} &\leq 0, & i \in \mathcal{N}. \quad (4.61)
\end{aligned}
$$

The objective function (4.43) expresses the NPV as the sum of the linear piece-wise approximations $y_i \approx f_i(s_i)$. The constraint sets (4.44)–(4.53) are exactly the same as the constraint sets (4.15)–(4.24) described above for the event-based formulation when considering the minimisation of makespan. Constraint sets (4.54) and (4.55) are introduced to determine s_i for each activity, by linking it with the event times relevant to each of the activities. A reasonable choice for the large number M in (4.54) and (4.55) is the latest possible finishing time of the schedule, *i.e.* $M = \mathcal{S}^{\mathcal{T}}_{|\mathcal{T}|}$.

Constraint sets (4.56)–(4.61) are duplicated from constraint sets (4.37)–(4.42) above, for the purpose of expressing the decision variables s_i and y_i as convex combinations of the piece-wise approximation (s_{iv}, y_{iv}), $v \in \mathcal{V}$.

4.4 Time-based costs and revenue

In the above formulations in pursuit of the maximisation of NPV, the parameter c_r is used to capture the unit cost or unit revenue for a resource $r \in \mathcal{R}$. In practice, however, costs are typically escalated into the future according to the *consumer price index* (CPI). The effect of applying the CPI to resource costs is that a different unit cost per time period is essentially obtained. This may easily be accommodated within the time-indexed formulation by adding a time index t to the cost parameter c_r. In the case of revenue drivers, such as mineral prices, escalation may also be considered to facilitate future mineral price scenarios. This may again be achieved by adding a time index to the specific revenue generating resource parameter c_r.

In this dissertation, however, the escalation of unit costs and revenues is not considered, although the mathematical formulations can easily be adapted to cater for it. The reason for this exclusion is due to the diverse interpretation of cost and revenue escalations in industry which make it difficult to incorporate in empirical studies.

4.5 Preprocessing and initial feasible solutions

The earliest and latest start times of an activity, denoted by $E_i^{\mathcal{N}}$ and $L_i^{\mathcal{N}}$, respectively, are functions of the precedence graph $H(\mathcal{N}, \mathcal{Z})$ and the scheduling horizon T of the RCSP. The latter is, however, dependent on the optimal solution of the RCSP. In the case of minimising the makespan, any feasible solution will provide an upper bound for T.

A heuristic is proposed below for the purpose of generating feasible solutions. These solutions, which will be feasible for all RCSP formulations defined above, may then be used to approximate the scheduling horizon T.

Algorithm 4.1: The time-indexed right-shift heuristic (TRSH)

```
N^B = ∅.
N^P = {0}.
s*_0 = 0.
for (0, j) ∈ Z do
  N^B = N^B ∪ {j}.
end for
t = 0
while N^B <> ∅ do
  period_feasible = false.
  for i ∈ N^B do
    if  φ(i, t, t)v_ir ≤ U_r − Σ_{i∈N^P} φ(i, t, s*_i)v_ir  ∀r ∈ R  then
      period_feasible = true.
    end if
  end for
  if  period_feasible = true  then
    Solve MKP(N^B, N^P)
    for i ∈ N^B do
      if  x_i = 1  then
        s*_i = S^T_t
        N^P = N^P ∪ {i}
        N^B = N^B \ {i}
        for (i, j) ∈ Z do
          N^B = N^B ∪ {j}.
        end for
      end if
    end for
  end if
  t=t+1
end while
Return upper bound T* = t and the solution vector s*.
```

4.5.1 The time-indexed right-shift heuristic (TRSH)

The concept behind the *time-indexed right-shift heuristic* (TRSH) is to start at time zero and progressively shift activities to the right (into the future), while maintaining feasibility with respect to the precedence graph and resource limits. The main challenge is to determine, for a current time period $t \in \mathcal{T}$, which activities should be right shifted and which ones should be selected to start. For this purpose the set $\mathcal{N}^B$ is introduced that contains activities eligible to begin in time period t and the set $\mathcal{N}^P$ is introduced that contains activities already processed and for which start time values s_i^* have been determined. Once an activity i is selected to start at a time period t, its start time is set to $s_i^* = S_t^{\mathcal{T}}$, it is added to $\mathcal{N}^P$ and it is removed from $\mathcal{N}^B$. An activity j may only be considered eligible to start in a time period t if all of its predecessors i have been completed prior to t. More specifically, $\mathcal{N}^B = \mathcal{N}^B \cup \{j\}$ only if $i \in \mathcal{N}^P$ for all $(i,j) \in \mathcal{Z}$, and only if $s_j \geq s_i + d_i$.

Selecting activities from the eligibility set $\mathcal{N}^B$ to start in time period t is achieved by solving a multi-knapsack problem. The items of the multi-knapsack problem are the activities in $\mathcal{N}^B$ and the constraints relate to the residual resource capacity available for time period t. The residual resource capacity for a resource $r \in \mathcal{R}$ is given by $U_r - \sum_{i \in \mathcal{N}^P} \phi(i, t, s_i^*) v_{ir}$, with s_i^* the start time that was assigned to each activity $i \in \mathcal{N}^P$.

Let $x_i \in \{0, 1\}$ denote the decision variable for knapsack item $i \in \mathcal{N}^B$ and let c_i be the associated cost coefficient. The approach to be followed for assigning appropriate values to each of the cost coefficients c_i follows below.

The objective of the multi-knapsack problem, MKP($\mathcal{N}^B$, $\mathcal{N}^P$), is to

$$\text{maximise} \sum_{i \in \mathcal{N}^B} c_i x_i, \tag{4.62}$$

subject to the constraints

$$\begin{aligned} s_j - s_i^* &\geq d_i, & i \in \mathcal{N}^P, j \in \mathcal{N}^B, \quad (4.63) \\ \sum_{i \in \mathcal{N}^B} v_{ir} x_i &\leq U_r - \sum_{i \in \mathcal{N}^P} \phi(i, t, s_i^*) v_{ir}, & r \in \mathcal{R}. \quad (4.64) \end{aligned}$$

The precedence constraints (4.63) ensure that an activity in the eligibility set $\mathcal{N}^B$ may only start once its predecessors in the processed set $\mathcal{N}^P$ have been completed. Constraints (4.64) are the residual resource capacity constraints.

The choice of c_i in the objective function (4.62) depends on whether the RCSP objective is to minimise makespan or to maximise NPV. In the case of makespan minimisation, a shortest path p_i is computed from each activity $i \in \mathcal{N}^B$ to the sink node based on the precedence graph $H(\mathcal{N}, \mathcal{Z})$. The edge weights for the graph are based on the durations of the activities. More specifically, the edge weight for an arc $(i, j) \in \mathcal{Z}$ is given by $w_{ij} = d_i$. The cost coefficients for MKP($\mathcal{N}^B$, $\mathcal{N}^P$) is then $c_i = 1/p_i$ in order to prefer activities that are associated with shorter total durations up to the sink activity.

In the case of NPV maximisation, the cost coefficients for MKP($\mathcal{N}^B$, $\mathcal{N}^P$) are $c_i = 1/d_i \sum_{r \in \mathcal{R}} c_r v_{ir}$ for each $i \in \mathcal{N}^B$. For this choice of cost coefficients activities are preferred for which the profit per time unit is a maximum. The complete algorithm of the TRSH, which incorporates the MKP($\mathcal{N}^B$, $\mathcal{N}^P$), is outlined in Algorithm 4.1. Note that the algorithm is initiated by adding the source node to the processed set, *i.e.* $\mathcal{N}^P = \{0\}$, and setting its start time to $s_0^* = 0$. Furthermore, the successors of the source node are then added to the eligibility set, *i.e.* $\mathcal{N}^B = \mathcal{N}^B \cup \{j\}$ for all $(0, j) \in \mathcal{Z}$.

The upper bound T^* for the scheduling horizon T is only valid for the RCSP when minimising makespan. In the case of maximising NPV, the estimate T^* may not be a valid upper bound, but the solution obtained from TRSH will still be feasible. In order to show that T^* may not be a valid upper bound obtained by TRSH, consider two independent activities that may be scheduled in parallel. If the resource capacity is sufficient, the two activities may start at the same time, which will then minimise the makespan. If one activity has a positive and the other a negative cost coefficient in the RCSP formulation, an optimal solution, which will maximise NPV, requires the activity with the negative cost coefficient to start as late as possible. Since the TRSH operates by assigning start times in a sequential fashion, a sub-optimal solution may be created which allows the two activities to be scheduled in parallel, resulting in a lower estimate of T. This will invalidate the optimal solution of starting the activity with a negative cost coefficient as late as possible.

4.5.2 Calculating earliest and latest starting times

Recall that $E_i^{\mathcal{N}}$ and $L_i^{\mathcal{N}}$ are the earliest and latest start times of an activity $i \in \mathcal{N}$, respectively. Conceptually, the approach towards determining $E_i^{\mathcal{N}}$ involves solving an optimisation problem in which the objective is to minimise the start time s_i of activity i, subject to the precedence constraints of the RCSP. Similarly, an optimisation problem that maximises the start time s_i of an activity i is solved to determine $L_i^{\mathcal{N}}$. It should be noted, however, that an upper bound T^* is required on s_i in order to prevent an unbounded solution in the case of solving the maximisation problem. The upper bound T^* may be provided by the solution of the TRSH.

The optimisation problem for determining $E_i^{\mathcal{N}}$ and $L_i^{\mathcal{N}}$ involves

$$\text{minimising / maximising} \quad s_i \tag{4.65}$$

subject to the constraints

$$\begin{aligned} s_j - s_i &\geq d_i, && (i,j) \in \mathcal{Z}, && (4.66) \\ s_i &\leq T, && i \in \mathcal{N}, && (4.67) \end{aligned}$$

for each activity $i \in \mathcal{N}$.

4.6 Summary

The underground mine scheduling problem, as a special case of the RCSP, entails determining the start times of mining activities in order to maximise NPV, while taking resource and mining-specific constraints into account. In this chapter, three mathematical formulations of the RCSP were provided for both the minimisation of makespan and the maximisation of NPV. Although the time-indexed formulation naturally accommodates the maximisation of NPV as a linear function, a new mathematical construct was proposed to facilitate the linearisation of the NPV objective function for both the resource flow-based and event-based formulations.

The time-indexed right-shift heuristic was proposed for the purpose of estimating the makespan T and for generating an initial feasible solution. Although all three of the RCSP formulations considered in this chapter benefits from a good estimate of the makespan T, it is an essential input to the time-indexed formulation, since the number of variables in the time-indexed formulation grows proportionately with an increase in T.

Although the three RCSP formulations presented in this chapter are different in terms of decision variables and constraints, all three formulations provide the same optimal solution when solved to optimality. There might, however, be a slight difference in objective function values due to the linearisation that is performed for the resource flow-based and event-based formulations.

Empirical tests involving the three RCSP formulations and the application of the time-indexed right-shift heuristic are presented in the next chapter. Although all three RCSP formulations are expected to provide the same optimal solution for a given problem instance, the computational properties of the formulations are investigated.

The acronyms TI, RF and EB are used in the remainder of this dissertation to refer to the time-indexed, the resource flow and the event-based formulations, respectively. From the context of the discussions to follow, it will be clear whether the minimisation or maximisation version of the said formulation is applicable.

CHAPTER 5

Model Selection through empirical evaluation

Contents

In this chapter, empirical tests are performed in order to investigate the computational properties of the three different RCSP formulations presented in the previous chapter.

The solution methodology adopted in this dissertation for solving the underground mine scheduling optimisation problem is set within an exact framework. The implication is that problem instances are either solved to optimality, or if the solution process is terminated prematurely with a feasible solution, a quantification of the degree to which optimality has been achieved is obtained. The primary measure of efficiency applied in this dissertation is, therefore, the number of problem instances from a given test set that is solved to optimality within a specified time limit. Secondary success indicators are the average time required to find an optimal solution and the average integrality gap in cases where instances cannot be solved to optimality but for which at least one feasible solution could be computed.

In the following section, details of existing problem instances found in the literature are provided as well as the characterisation of problem instances according to various tractability indicators. Alternative problem instances are also proposed that will better reflect characteristics of typically underground mine scheduling optimisation problems.

5.1 Test instance data

Several data sets for the RCSP and its variants are available in the academic research community for the purpose of testing model efficiency and algorithmic ideas. For instance, the *project scheduling problem library* (PSPLIB) [80] is a repository of RCSP problem instances that has been referenced extensively over the years. The PSPLIB comprises the data sets J30, J60, J90 and J120, which are sets of RCSP instances each containing 30, 60, 90 and 120 activities, respectively. Each data set consists of 480 different problem instances, except for the J120 data set which contains 600 problem instances. Details on how these problem instances were created can be found in Kolish and Sprecher [53]. Other well-known data sets are the 39 problem instances of Baptiste and Pape [11] and the instances established by Carlier and Néron [20].

An important collective contribution by the research community has been the characterisation of problem instances according to various indicators. Some of the indicators may be used to distinguish between "easy" and "hard" instances and are briefly described below (see [2] for details):

Order strength (OS) is a measure of parallelism of the underlying precedence graph. That is, a problem instance for which $OS = 0$ indicates that all activities occur in parallel, whereas $OS = 1$ indicates that all activities are ordered in series. The hardness of problem instances increases with a decrease in the value of OS.

Network complexity (NC) is the average number of incident arcs per node in the precedence graph. Higher levels of NC are associated with harder problem instances.

Resource factor (RF) measures the average number of resources required per activity. It has been observed empirically that RF increases with an increase in the hardness of problem instances.

Resource strength (RS) is a measure which combines resource requirements per activity and peak resource demand due to a precedence-feasible schedule based on the earliest start times of activities. Problem instances for RS close to zero are considered much harder than problem instances for which RS is close to one.

Disjunction ratio (DR) provides an indication of how many activities may be scheduled in parallel by taking resource requirements and precedence relations into account. Highly disjunctive problem instances are considered to be easier than cumulative instances that have a lower disjunction ratio.

For the purpose of this empirical evaluation, problem instances from the J30, J60 and J90 data sets were used, as well as the Baptiste and Le Pape (BL), and the Carlier and Néron (CN) problem instances. Due to the large number of different algorithmic options that had to be examined, a sample of 50 instances were taken from each of the data sets J30, J60 and J90. As criteria for selecting the samples the parameters RS and DR were calculated for each problem instance. It has been shown empirically that problem instances with low RS and DR indicators are "harder" to solve than problem instances with higher RS and DR values (Baptiste and Pape [11]). The instances in the J30, J60 and J90 data sets were ranked according to the average of their RS and DR values (both within the interval $[0, 1]$) and the bottom 50 instances, according to this ranking, were selected as the sample in each case. That is, 50 of the "harder" problem instances were obtained for each of the J30, J60 and J90 data sets. The newly created data sets are henceforth called the J30(50), J60(50) and the J90(50) data sets.

Table 5.1 contains statistics on the indicators calculated for the existing problem instances from the literature as well as the sampled data sets J30(50), J60(50) and J90(50). It is evident from these statistics that the sampling approach applied to obtain a sample with "harder" problem instances for the J30, J60 and J90 data sets, delivered the desired results. The average RS and DR values for the sampled data sets J60(50) and J90(50), are lower than the average RS and DR values of the original data sets J60 and J90. Although the average DR for the J30(50) data set is slightly more than the DR of the original data set J30, it still has a much lower average RS. Furthermore, although no data set in Table 5.1 dominates with respect to all of

Data set	number of activities	avg. activity duration	*OS* avg	*OS* σ	*NC* avg	*NC* σ	*RF* avg	*RF* σ	*RS* avg	*RS* σ	*DR* avg	*DR* σ
J30	30	5.1	0.52	0.09	1.81	0.26	0.63	0.28	0.62	0.29	0.56	0.11
J30(50)	30	5.1	0.45	0.07	1.64	0.21	0.59	0.29	0.2	0.02	0.57	0.07
J60	60	5.3	0.4	0.08	1.81	0.25	0.63	0.28	0.6	0.29	0.41	0.09
J60(50)	60	5.3	0.32	0.04	1.58	0.13	0.64	0.3	0.2	0.01	0.35	0.05
J90	90	5.4	0.34	0.08	1.8	0.25	0.63	0.28	0.60	0.29	0.34	0.08
J90(50)	90	5.4	0.26	0.03	1.6	0.12	0.66	0.27	0.2	0.01	0.27	0.04
BL	22–27	2.8	0.34	0.07	1.67	0.13	0.66	0.06	0.34	0.09	0.34	0.07
CN	17–35	6.2	0.23	0.07	1.61	0.06	1	0	0.17	0.08	0.44	0.19

TABLE 5.1: *Tractability indicators for the existing problem instances from the literature.*

Data set	number of activities	avg. activity duration	*OS* avg	*OS* σ	*NC* avg	*NC* σ	*RF* avg	*RF* σ	*RS* avg	*RS* σ	*DR* avg	*DR* σ
J30(50)D	30	252.7	0.45	0.07	1.6	0.21	0.59	0.29	0.25	0.09	0.57	0.07
J60(50)D	60	257.8	0.32	0.04	1.6	0.13	0.64	0.3	0.22	0.04	0.35	0.05
J90(50)D	90	264.7	0.26	0.04	1.6	0.12	0.66	0.27	0.21	0.03	0.27	0.04
BLD	22–27	140.0	0.34	0.07	2.93	0.46	0.66	0.06	0.23	0.06	0.34	0.07
CND	17–35	309.3	0.23	0.06	1.8	0.27	0.99	0.06	0.07	0.08	0.44	0.18

TABLE 5.2: *Tractability indicators for the randomly generated problem instances with longer activity durations.*

the tractability indicators, the J90(50) data set is dominant when considering the RS and DR indicators.

Note that the average activity durations for the data sets listed in Table 5.1 are all relatively short. In order to evaluate the efficiency of different RCSP formulations within the context of underground mining, problem instances with longer activity durations need to be considered. For this purpose new problem instances were created by increasing the activity durations of the existing problem instances. The duration d_i for each activity $i \in \mathcal{N}$ was adjusted to $d_i = d_i \times \tilde{d}$, with $\tilde{d} \sim U(1, 100)$. The newly created data sets with longer durations are called the J30(50)D, J60(50)D, J90(50)D, CND and the BLD data sets. Statistics on the indicators for these data sets are listed in Table 5.2. The tractability indicators for the newly created problem instances with longer activity durations are not much different from those of the original problem instances, and once again, the J90(50)D data set is dominant when considering the RS and DR indicators.

5.2 Computational results

All of the empirical tests reported in this dissertation were performed on an HP Compaq Elite 8300 processor with four cores, 32GB of RAM and operating at a speed of 3.4 GHz. SuSE Linux was used as operating system and the IBM product, CPLEX v12.6 [48], was used as MILP solver.

The time limit for each computational run was determined as a function of the size of the problem instance being solved. More specifically, 10 seconds of computing time was allowed for each activity in the problem instance. As an example, for the J30(50) data set in which each problem instance comprises 30 activities, the run time was limited to 300 seconds.

Two performance indicators are used for the purpose of reporting results. The first indicator is the percentage of instances in the data set for which an optimal solution could be computed in the allocated time. The average computing times recorded to obtain optimal solutions are also provided. The second indicator is the percentage of instances in the data set for which at least

		Solved to optimality		Feasible Solutions	
Data set	Model	Instances (%)	Time (s)	Instances (%)	Gap (%)
J30(50)	TI-MIN	62	38.1	100	7.3
	RF-MIN+(4.11)	36	15.3	100	20.6
	RF-MIN	34	3.3	100	20.8
	RF-MIN+(4.11)+(4.12)	34	21	100	27.2
	RF-MIN+(4.12)	34	25.7	100	28.3
	EB-MIN	2	10.4	100	28.8
J60(50)	TI-MIN	28	44.1	100	33.2
	RF-MIN+(4.11)+(4.12)	10	79.3	80	65.6
	RF-MIN	10	119.8	100	34.3
	RF-MIN+(4.11)	0	—	100	33.6
	EB-MIN	0	—	100	49.2
	RF-MIN+(4.12)	0	—	18	44.5
J90(50)	TI-MIN	20	249.4	100	33.2
	RF-MIN+(4.11)	2	665.5	68	84.4
	RF-MIN	0	—	100	45.7
	RF-MIN+(4.11)+(4.12)	0	—	100	49.1
	EB-MIN	0	—	100	56.1
	RF-MIN+(4.12)	0	—	96	44.2
BL	TI-MIN	100	1.5	100	0
	RF-MIN+(4.11)	15.4	95.2	100	19.9
	RF-MIN	12.8	116.1	100	19.6
	RF-MIN+(4.12)	10.3	73.2	100	21.4
	RF-MIN+(4.11)+(4.12)	10.3	116.7	100	20.6
	EB-MIN	0	—	100	24.4
CN	TI-MIN	94.6	41.5	100	0.2
	RF-MIN+(4.11)	0	—	100	59.8
	RF-MIN	0	—	100	59.9
	EB-MIN	0	—	100	60.2
	RF-MIN+(4.11)+(4.12)	0	—	100	62.7
	RF-MIN+(4.12)	0	—	100	63

TABLE 5.3: *The computational efficiency of the RCSP formulations in respect of standard data sets from the literature.*

one feasible solution could be computed within the specified time limit. The average gap is also reported for all of the problem instances for which feasible solutions could be computed.

The results in subsequent tables are first ranked according to the percentage of instances for which optimal solutions could be computed, then according to the average computing time it took to compute the optimal solutions, the percentage of instances for which feasible solutions were obtained, and finally, according to the average integrality gap obtained for all feasible solutions computed.

5.2.1 Minimisation of makespan

The results obtained for the standard RCSP data sets from literature are presented in Table 5.3. The main purpose of presenting these results is to evaluate the computational efficiency of the various model formulations from Chapter 4, and specifically, to evaluate the effect on computing times when the valid inequalities (4.11) and (4.12) are included in the RF formulation.

From the results in Table 5.3 it is evident that the TI formulation outperformed all other formulations in terms of both the number of instances solved to optimality and the average gap

		Solved to optimality		Feasible Solutions	
Data set	Model	Instances (%)	Time (s)	Instances (%)	Gap (%)
J30(50)D	RF-MIN+(4.11)	38	5.4	100	19
	RF-MIN	38	9.4	100	19
	RF-MIN+(4.11)+(4.12)	36	25.5	100	24.2
	RF-MIN+(4.12)	34	15	100	26.2
	EB-MIN	6	101.9	100	24.4
	TI-MIN	0	—	4	87.1
J60(50)D	RF-MIN	6	60.5	100	31.2
	RF-MIN+(4.11)	4	2.2	100	32.5
	EB-MIN	2	500.6	100	48.7
	RF-MIN+(4.11)+(4.12)	0	—	82	65.7
	RF-MIN+(4.12)	0	—	12	39.2
	TI-MIN	0	—	0	—
J90(50)D	RF-MIN+(4.11)	2	185.7	96	42.5
	RF-MIN	2	369.2	96	41.2
	EB-MIN	0	—	100	61.6
	RF-MIN+(4.11)+(4.12)	0	—	72	82.8
	RF-MIN+(4.12)	0	—	0	—
	TI-MIN	0	—	0	—
BLD	RF-MIN+(4.11)+(4.12)	30.8	49.1	100	12.9
	RF-MIN+(4.12)	30.8	64	100	13.3
	RF-MIN+(4.11)	28.2	15.8	100	12
	RF-MIN	28.2	15.8	100	12.2
	EB-MIN	17.9	38.2	100	14.2
	TI-MIN	5.1	80.3	100	64.2
CND	RF-MIN	0	—	100	55.3
	RF-MIN+(4.11)	0	—	100	55.3
	EB-MIN	0	—	100	55.5
	RF-MIN+(4.11)+(4.12)	0	—	100	58.9
	RF-MIN+(4.12)	0	—	98.2	59.4
	TI-MIN	0	—	41.1	99

TABLE 5.4: *The computational efficiency of the RCSP formulations in respect of newly created data sets with extended activity durations.*

obtained. More specifically, all of the BL instances and 94.6% of the CN instances were solved to optimality with the TI formulation. The worst performing formulation appears to be the EB formulation with only 2% of the J30(50) problem instances solved to optimality.

A marginal improvement in computing time is observed when the valid inequalities (4.11) are included in the RF formulation. Interestingly, in almost all cases, except for the J60(50) data set, the inclusion of (4.12) and the combination of (4.11) and (4.12) in the RF formulation, has a negative effect on computing times.

Use of the TI formulation proves to be ineffective when considering problem instances with longer activity durations, as indicated by Table 5.4. By employing the TI formulation, optimal solutions could be computed for only the BLD problem instances. It is disappointing to note that no feasible solutions could be computed for the J60(50)D and the J90(50)D problem instances. The poor performance of the TI formulation is nevertheless to be expected since an increase in activity duration results in a proportional increase in the number of binary variables found in the TI formulation. The results obtained when solving the RF formulation are promising and the addition of the valid inequalities (4.11) brings about only a moderate improvement in computing times. The inclusion of both (4.11) and (4.12) in the RF formulation again have a negative effect on computing times for all of the problem instances, except for the BLD data set. The benefit of including (4.11) and (4.12) in the RF formulation for solving the BLD problem instances is, however, marginal and is overshadowed by the poor performance observed for both the J60(50)D and J90(50)D data sets. More specifically, by including (4.11) and (4.12) in the RF formulation, feasible solutions could be computed for only 82% of the J60(50)D and 72% of the J90(50)D problem instances.

The poor performance of the valid inequalities (4.11) and (4.12) justifies the adoption of the RF formulation for solving the RCSP, without including these valid inequalities. All subsequent results are, therefore, reported for the RF formulation from which the valid inequalities (4.11) and (4.12) have been excluded.

The benefits of applying the time-indexed right-shift heuristic (TRSH), presented in Chapter 4, are reported in Table 5.5. There are two ways in which the TRSH may be employed — first by solving the TRSH and only using the upper bound T on the scheduling horizon to calculate the latest start time of each activity. The abbreviation TRSH-EST is used in Table 5.5 to indicate the use of the estimated upper bound T in the formulation of the RCSP. Secondly, both the estimated upper bound T and the feasible solution generated by the TRSH may be used. In this case the abbreviation TRSH-FEAS is used.

For the J30(50)D data set there is no real benefit in using TRSH in conjunction with the RF or the EB formulation. Significant improvements were observed, however, for the TI formulation when applying TRSH. Without the heuristic, no optimal solutions could be computed when applying the TI formulation and by using TRSH-FEAS, 16% of the J30(50)D instances were solved to optimality. In the case of the J60(50)D and the J90(50)D data sets, improvements were observed when applying TRSH in conjunction with both the RF and the TI formulation. Although only two percent of the J90(50)D problem instances could be solved with and without the TRSH, the average solution time improved from 369.2 seconds to 40.3 seconds when applying TRS-FEAS in conjunction with the RF formulation. No real benefit was observed, however, for the BLD and CND data sets when using TRSH.

The overall impression is that the best results are obtained when adopting the RF formulation if considering the minimisation of makespan. This appears to be true for both the standard problem instances from the literature as well as for the newly created instances that have longer activity durations. Further improvements are possible when applying TRSH to estimate the scheduling horizon and to generate an initial feasible solution. Although improvements are possible when applying TRSH with the TI formulation, results that will be presented below for the maximisation of NPV discourage adoption of the TI formulation. The main reason for this is that the TRSH cannot be used to estimate a valid upper bound for the scheduling horizon when considering the maximisation of NPV (see discussion in Section 4.5.1). It is evident from the results above that the TI formulation is not a viable option without an upper bound T.

5.2.2 Maximisation of NPV

Computational results for the maximisation of NPV are presented in Table 5.6. The RF and EB formulations were applied to the newly created data sets having problem instances with longer activity durations. Since no valid upper bound for T can be estimated by means of the TRSH, only the feasible solutions generated by the TRSH are used as initial feasible solutions to the RF and EB formulations.

Positive results may be reported for the case where TRSH is used in conjunction with the RF formulation in all of the problem instances listed in Table 5.6. Although the percentage of

			Solved to optimality		Feasible Solutions	
Data set	Model	Heuristic	Instances (%)	Time (s)	Instances (%)	Gap (%)
J30(50)D	RF-MIN	TRSH-FEAS	38	7.5	100	18.7
	RF-MIN	—	38	9.4	100	19
	RF-MIN	TRSH-EST	38	11.7	92	17.3
	TI-MIN	TRSH-FEAS	16	111.7	94	67.5
	TI-MIN	TRSH-EST	14	140	48	40
	EB-MIN	TRSH-EST	6	100.8	88	23.4
	EB-MIN	—	6	101.9	100	24.4
	EB-MIN	TRSH-FEAS	4	8.5	100	24.6
	TI-MIN	—	0	—	4	87.1
J60(50)D	RF-MIN	TRSH-FEAS	8	123.9	100	29.8
	RF-MIN	—	6	60.5	100	31.2
	RF-MIN	TRSH-EST	6	99	34	9.6
	TI-MIN	TRSH-EST	2	110	8	10.7
	TI-MIN	TRSH-FEAS	2	518.7	52	60.2
	EB-MIN	TRSH-FEAS	0	—	100	38.8
	EB-MIN	TRSH-EST	0	—	8	48.5
	EB-MIN	—	0	—	100	48.7
	TI-MIN	—	0	—	0	—
J90(50)D	RF-MIN	TRSH-FEAS	2	40.3	100	35.6
	RF-MIN	—	2	369.2	96	41.2
	RF-MIN	TRSH-EST	2	811.6	20	8.7
	EB-MIN	—	0	—	100	61.6
	EB-MIN	TRSH-FEAS	0	—	98	42.1
	TI-MIN	TRSH-FEAS	0	—	26	46.8
	TI-MIN	—	0	—	0	—
	EB-MIN	TRSH-EST	0	—	0	—
	TI-MIN	TRSH-EST	0	—	0	—
BLD	RF-MIN	TRSH-FEAS	28.2	14.8	100	12.1
	RF-MIN	—	28.2	15.8	100	12.2
	RF-MIN	TRSH-EST	28.2	22.3	100	12
	EB-MIN	—	17.9	38.2	100	14.2
	EB-MIN	TRSH-FEAS	17.9	60.3	100	14.3
	EB-MIN	TRSH-EST	12.8	66.1	100	14.2
	TI-MIN	TRSH-FEAS	12.8	75.7	100	23.9
	TI-MIN	TRSH-EST	10.3	51.6	100	23.8
	TI-MIN	—	5.1	80.3	100	64.2
CND	RF-MIN	—	0	—	100	55.3
	EB-MIN	—	0	—	100	55.5
	EB-MIN	TRSH-FEAS	0	—	100	55.6
	RF-MIN	TRSH-FEAS	0	—	100	55.6
	EB-MIN	TRSH-EST	0	—	96.4	55.3
	TI-MIN	TRSH-FEAS	0	—	80.4	90.5
	RF-MIN	TRSH-EST	0	—	73.2	51.8
	TI-MIN	TRSH-EST	0	—	50	90.4
	TI-MIN	—	0	—	41.1	99

TABLE 5.5: *The computational efficiency of the RCSP formulations in respect of the newly created data sets with extended activity durations.*

optimal solutions found for the J90(50)D does not improve when applying the TRSH in the case of the RF formulation, it does, however, provide feasible solutions to all problem instances with an average integrality gap of only 10.2%.

Less favourable results were observed for the EB formulation — no optimal solutions could be

			Solved to optimality		Feasible Solutions	
Data set	Model	Heuristic	Instances (%)	Time (s)	Instances (%)	Gap (%)
J30(50)D	RF-MAX	TRSH	72	44.4	96	0.5
	RF-MAX	—	66	39.1	92	0.5
	EB-MAX	TRSH	0	—	100	70.2
	EB-MAX	—	0	—	100	70.5
J60(50)D	RF-MAX	TRSH	32	114.3	100	7.8
	RF-MAX	—	26	84.5	98	9.6
	EB-MAX	TRSH	0	—	100	75.2
	EB-MAX	—	0	—	100	99.6
J90(50)D	RF-MAX	—	4	66.2	72	17.7
	RF-MAX	TRSH	4	85.7	100	10.2
	EB-MAX	TRSH	0	—	100	45.3
	EB-MAX	—	0	—	100	62.8
BLD	RF-MAX	TRSH	92.3	7.4	100	0.1
	RF-MAX	—	92.3	9	100	0.1
	EB-MAX	TRSH	0	—	100	53.4
	EB-MAX	—	0	—	97.4	54.3
CND	RF-MAX	TRSH	16.1	64.3	96.4	6.5
	RF-MAX	—	14.3	59.8	94.6	7.1
	EB-MAX	TRSH	0	—	98.2	22.8
	EB-MAX	—	0	—	96.4	23.1

TABLE 5.6: *The computational efficiency of the RCSP formulations when maximising NPV in respect of the newly created data sets with extended activity durations.*

computed for any of the problem instances in Table 5.6. The EB formulation did, however, provide feasible solutions to all of the problem instances, except for the CND data set. The application of the TRSH resulted in an improvement in the integrality gap for most of the problem instances when applying the EB formulation.

5.3 Summary

The emphasis in this dissertation is on the adoption of an exact solution framework for solving the RCSP. The implication is that the method for generating solutions is separated from the problem formulation. The benefit of this is that generic constraints may be added to RCSP formulation in order to cater for other mining-specific requirements that may not have been addressed in this study.

Empirical results involving the three RCSP formulations from Chapter 4 were presented in this chapter. Details on the characterisation of existing problem instances found in the literature were provided and additional problem instances were proposed which better reflect characteristics of typical underground mine scheduling optimisation problems.

The assessment of which problem formulation is most suited was based on two criteria — the percentage of instances in the data set for which an optimal solution could be computed and the percentage of instances in the data set for which at least one feasible solution could be computed within the specified time limit.

The most promising RCSP formulation appears to be the RF formulation, specifically for the newly created problem instances having longer activity durations and for the objective function that involves the maximisation of NPV. The formulation of choice for solving the underground mine scheduling optimisation problem is, therefore, the RF formulation.

It is acknowledged that although promising results were reported for the RF formulation, based on randomly generated data sets, the use of real-world underground mine scheduling problem instances may pose more of a challenge to solve. A distinctive feature of underground mine scheduling problem instances are the large number of resource capacities which have to be taken into account. Several algorithmic enhancements are therefore proposed in the next chapter to improve computational efficiency, by explicitly taking into account that problem instances may contain a large number of resources.

CHAPTER 6

Algorithmic improvements

Contents

The computational complexity of the RCSP has been established as being $\mathcal{NP}$-hard (Blazewicz *et al.* [14]). It is, therefore, unlikely that an algorithm can be developed to solve the underground mine scheduling optimisation problem to optimality in reasonable time for realistically sized problem instances. A typical course of action is either to improve lower bounds through stronger problem formulations and the application of valid inequalities, or to improve the generation of primal solutions through the use of heuristics. Depending on the problem formulation adopted, an alternative is to make use of a decomposition approach for speeding up computations.

As a first attempt to improve computational efficiency when solving the RCSP according to the RF formulation, a variable and constraint reduction approach are suggested. Variable reduction is achieved by reducing the number of arcs in the resource flow graph $G(\mathcal{N}, \mathcal{A})$ and constraint reduction is achieved by aggregating some of the constraints in the RF formulation, without removing feasible solutions from the problem space.

The final contribution of this chapter is a Benders decomposition solution approach which allows the mine scheduling optimisation problem to be solved within a branch-and-cut framework. The motivation for exploring the use of decomposition is to improve scalability, specifically when problem instances with a large number of resources are considered. A separation procedure and a primal heuristic are also proposed as part of the Benders decomposition approach.

6.1 Variable and constraint reduction

An alternative formulation of some of the constraints in the RF formulation of the RCSP is described in this section. Although the resulting reformulation may possibly be weaker than the original, it may be computationally more efficient since it employs fewer variables and constraints to describe the same feasible region. The anticipated advantage is that there is less of a burden on the LP solver when solving the sub-problems of the branch-and-bound tree. In addition, a graph reduction approach is described, which may also result in the RF formulation having fewer variables and constraints.

6.1.1 An aggregated constraint formulation (AC)

Consider the constraint sets (4.10) and (4.36),

$$f_{ijr} - \min\{v_{ir}/d_i, v_{jr}/d_j\}z_{ij} \quad \leq \quad 0, \;\; (i,j) \in \mathcal{A}, \;\; r \in \mathcal{R}, \tag{6.1}$$

in the RF-MIN and RF-MAX formulations, respectively. These constraints are responsible for allowing the flow of resources whenever the linear ordering variable z_{ij} has been set to one. There are in total $|\mathcal{A}| \times |\mathcal{R}|$ of these constraints in both the RF-MIN and RF-MAX problem formulations, and an increase in the number of resources in the problem instance being solved may have a negative effect on computational efficiency. Since the linear ordering variables z_{ij} are not indexed by a resource, it is, however, possible to rewrite these constraints in the aggregated form

$$\sum_{r\in\mathcal{R}} f_{ijr} - \left(\sum_{r\in\mathcal{R}} U_r\right) z_{ij} \;\leq\; 0, \;\; (i,j) \in \mathcal{A}. \tag{6.2}$$

Both constraint sets (6.1) and (6.2) are based on a big-M formulation, which is considered to be an undesirable property. The latter constraint set, however, clearly comprises fewer constraints since the individual resource flows f_{ijr} are aggregated over all the resources $r \in \mathcal{R}$. Although the aggregated version (6.2) is considered to be theoretically weaker than (6.1), some practical benefit is expected in terms of computing times, especially in the case where a large number of different resources is considered.

6.1.2 Graph reduction (GR)

The approach proposed in this section to reduce the resource flow graph is inspired by the observation that resource flows are consistent with precedence relationships. More specifically, it is evident from the linear ordering constraints that for an activity i, which is a predecessor of j (*i.e.* $s_i + d_i \leq s_j$), resources are not allowed to flow from j to i. Applying this recursively, no flows are allowed from j to any of its ancestors, as defined by the precedence graph $H(\mathcal{N}, \mathcal{Z})$. Based on this observation, some of the variables and constraints in the original problem formulations may be ignored by considering a resource flow subgraph $G'(\mathcal{N}, \mathcal{A}')$, with $\mathcal{A}' \subset \mathcal{A}$ constructed in such a manner that there are no arcs from a node i to any of its ancestors.

The set of predecessor activities of activity $i \in \mathcal{N}$ is defined as $\mathcal{P}(i) \subseteq \mathcal{N}$. These predecessor activities are the immediate predecessors of activity i. For each of the predecessors $j \in \mathcal{P}(i)$, a predecessor list $\mathcal{P}(j)$ exists. Continuing in a recursive manner, all of the activities along each possible path from node i to a root node of the precedence graph is obtained. This set of activities is denoted by $\mathcal{P}^+(i) \subseteq \mathcal{N}$ and denotes the set of ancestors for activity i.

The set of arcs of the reduced resource flow graph $G'(\mathcal{N}, \mathcal{A}')$ is defined as $\mathcal{A}' = \{(i,j) : j \notin \mathcal{P}^+(i)\}$. By replacing the set $\mathcal{A}$ in the RF-MIN formulation (4.5)–(4.10) and in the RF-MAX formulation (4.31)–(4.42) by the set $\mathcal{A}'$, graph reduction is applied which results in fewer variables and constraints in the RF formulation.

6.2 A branch-and-cut solution framework

The *branch-and-cut* framework is an adaptation of the branch-and-bound approach and it has been applied successfully in the solution of many large-scale optimisation problems (Padberg and Rinaldi [78]). The framework entails relaxing some of the constraints in the original problem and only incorporating them back into the branch-and-bound procedure if needed. In order to illustrate the concept more concretely, an overview of the branch-and-bound method is provided. A discussion then follows on the implementation of a Benders decomposition approach within the context of the branch-and-cut framework.

6.2.1 Branch-and-bound basics

The branch-and-bound method is based on the notion of a divide-and-conquer approach (Land and Doig [59]) and involves the iterative solution of LP sub-problems. That is, the feasible region of the MILP problem to be solved is systematically partitioned into smaller regions based on the solutions obtained by solving the LP sub-problems. To illustrate the process, let $\boldsymbol{x}$ denote the vector of integer decision variables in the formulation of an MILP problem. Let LP_0 denote the first LP sub-problem obtained by relaxing the integer requirements on $\boldsymbol{x}$. If LP_0 is infeasible, the original MILP problem is also infeasible. If, on the other hand, the solution $\boldsymbol{x}^0$ to LP_0 satisfies the integer requirements for all of the variables in $\boldsymbol{x}$, an optimal solution has been found. Otherwise, at least one of the components of $\boldsymbol{x}$ is non-integral and, assuming that the MILP problem is a minimisation problem, the current lower bound $\underline{Z}^*$ is assigned the objective function value Z_0 obtained by solving the sub-problem LP_0. The initial upper bound, again assuming that the MILP problem is a minimisation problem, is taken as $\overline{Z}^* = \infty$. Partitioning of the feasible region of the MILP problem is now achieved by creating two sub-problems LP_1 and LP_2. If x_k is one of the components in $\boldsymbol{x}$ that does not have an integer solution when solving LP_0, then LP_1 is obtained by adding the constraint $x_k \leq \lfloor x_k^0 \rfloor$ to the parent problem LP_0, where x_k^0 is the solution of the k-th component of $\boldsymbol{x}$. Similarly, the sub-problem LP_2 is obtained by adding the constraint $x_k \geq \lfloor x_k^0 \rfloor$ to the parent problem LP_0. This step in the branch-and-bound process is called *branching*. It should be noted that in the case where more than one component of $\boldsymbol{x}$ is non-integral, some rule has to be applied in order to select the variable on which branching is performed. Commercial MILP solvers typically provide several branching strategies from which to choose. The computational study by Linderoth and Savelsbergh [64] showed, however, that no branching strategy dominates and that the effect of a particular strategy depends on the specific problem instance being solved.

If, during the branch-and-bound process, a solution to a sub-problem p is found to be integral, it becomes the *incumbent integer solution*, provided that the objective function value Z_p of the corresponding sub-problem is less than the current upper bound $\overline{Z}^*$. If this is the case, the upper bound is updated by setting $\overline{Z}^* = Z_p$. Otherwise, if the sub-problem is not integral and its objective function is still less than the current upper bound $\overline{Z}^*$, it is added to an open list of sub-problems. A sub-problem may be *pruned*, and consequently not be added to the open list of sub-problems, whenever it is either infeasible or its objective function value exceeds the current upper bound $\overline{Z}^*$.

It is convenient to represent the progression of the branch-and-bound process by means of a binary tree, with the nodes of the tree corresponding to each of the sub-problems. The sub-problems in the open list are referred to as *dangling nodes* and the process of selecting the next sub-problem for branching is called *node selection*. Similar to different branching strategies, different criteria exist for selecting a node from the open list. Nodes are removed from the open list if they are selected for branching or if their objective function values exceed that of a newly found incumbent integer solution.

6.2.2 Cut generation in the branch-and-bound

The branch-and-cut approach provides a generic framework in which a relaxed version of an MILP problem is solved using the branch-and-bound method. Feasibility with respect to the original problem is achieved through the dynamic generation of constraints, which are called *cuts*. Consider a sub-problem p being solved as part of the branch-and-bound process. The current solution $\boldsymbol{x}^p$ is tested for feasibility against the constraints of the original MILP problem. If $\boldsymbol{x}^p$ is found to be infeasible, a cut is derived and added to the current sub-problem. The sub-problem is repeatedly solved until no more violated cuts can be found. The simplest implementation of such a process is to treat the relaxed constraints as *lazy constraints*. That is, each lazy constraint is explicitly tested for satisfaction in each of the sub-problems of the branch-and-bound approach. The benefit in terms of computing times may, however, by negated when a large number of lazy constraints are present.

An alternative to using lazy constraints is to generate cuts implicitly through the use of *separation algorithms*. That is, given an existing solution $\boldsymbol{x}^p$ at a node of the branch-and-bound approach, a separation algorithm is applied to generate a cut using information from $\boldsymbol{x}^p$. The cut is then appended to the relaxed MILP formulation to cut off the current infeasibility. Figure 6.1 contains a flow diagram of the branch-and-cut process in which the application of a separation algorithm for cut generation as part of the branch-and-bound approach is illustrated.

The choice of constraints to relax is driven in most cases by the model complexity. For instance, the resource flow-based formulation of the RCSP grows exponentially in the number of variables with an increase in the number of activities. Intuitively, by relaxing all the resource flow-related constraints, the computational burden should be less when solving the branch-and-bound sub-problems. The challenge, however, is to find a suitable separation algorithm which is able to find violated cuts for the remaining MILP problem.

6.2.3 The resource flow Benders master problem (RFM)

The suggested approach for solving the resource flow RCSP within a branch-and-cut framework is to make use of Benders decomposition [12]. The resource flow variables and their associated constraints are projected out of the original MILP formulation and a separation algorithm is presented below which is responsible for generating cuts so as to ensure feasibility.

Consider the RF-MIN formulation (4.5)–(4.10) in Chapter 4. The variables in the RF formu-

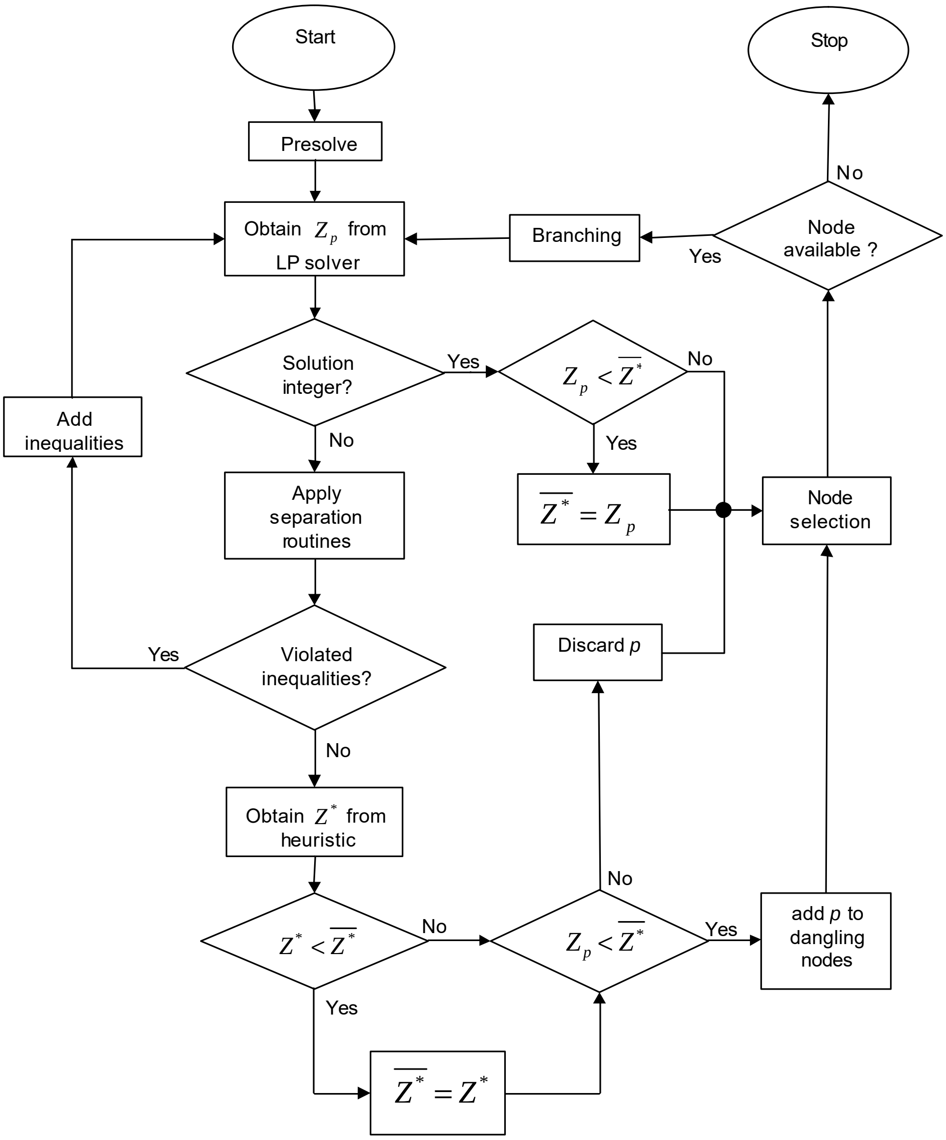


FIGURE 6.1: *Outline of the branch-and-cut framework.*

lation are the start time variables $\boldsymbol{s} \in \mathbb{R}_+^{|\mathcal{N}|}$, the linear ordering variables $\boldsymbol{z} \in \{0,1\}^{|\mathcal{A}|}$ and the resource flow variables $\boldsymbol{f} \in \mathbb{R}_+^{|\mathcal{A}|\times|\mathcal{R}|}$. For illustration purposes, let $\boldsymbol{x} \in \mathbb{R}_+^{|\mathcal{N}|} \times \{0,1\}^{|\mathcal{A}|}$ denote the adjoint vectors $(\boldsymbol{s}, \boldsymbol{z})$ and let $\mathcal{X}$ denote the polyhedron induced by the constraint sets (4.6)–(4.7) in the RF-MIN formulation. Let A and B be appropriate coefficient matrices associated with $\boldsymbol{x}$ and $\boldsymbol{f}$, respectively, which correspond to the coefficients in the RF-MIN constraints (4.9)–(4.10). Furthermore, let $\boldsymbol{b}$ correspond to the right-hand side of the same constraints and let $\boldsymbol{c}$ and $\boldsymbol{d}$ be the cost vectors associated with $\boldsymbol{x}$ and $\boldsymbol{f}$, respectively. It is evident from the RF-MIN objective function (4.5) that all entries of $\boldsymbol{c}$ are zero except for the component corresponding to the sink activity s_N. Consequently, all entries of the cost vector $\boldsymbol{d}$ are also zero since there are no costs associated with the flow variables $\boldsymbol{f}$ in the objective function.

The objective of the RF-MIN, in matrix form, is to

$$\min \boldsymbol{c}^T \boldsymbol{x} + \boldsymbol{d}^T \boldsymbol{f}, \tag{6.3}$$

$$\text{s.t. } A\boldsymbol{x} + B\boldsymbol{f} \leq \boldsymbol{b}, \tag{6.4}$$

$$\boldsymbol{x} \in \mathcal{X}. \tag{6.5}$$

Since no flow variables are present in the polyhedron $\mathcal{X}$, the above problem can be decomposed into a master problem

$$\min \boldsymbol{c}^T \boldsymbol{x}, \tag{6.6}$$

$$\boldsymbol{x} \in \mathcal{X} \tag{6.7}$$

and a sub-problem

$$\min \boldsymbol{d}^T \boldsymbol{f}, \tag{6.8}$$

$$\text{s.t. } B\boldsymbol{f} \leq \boldsymbol{b} - A\boldsymbol{x}^*, \tag{6.9}$$

where $\boldsymbol{x}^*$ is the solution obtained by solving the master problem.

The master problem (6.6)–(6.7) is an MILP problem and may be solved using the branch-and-bound method. The sub-problem (6.8)–(6.9) is an LP problem, given a fixed $\boldsymbol{x}^*$. The outcome of solving the LP sub-problem is that either an optimal solution $\boldsymbol{f}^*$ is obtained, or the problem is infeasible as a result of $\boldsymbol{x}^*$. In the case where the sub-problem is infeasible, a cut $(\boldsymbol{b}-A\boldsymbol{x}^*)^T\boldsymbol{w} \geq 0$ can be derived where $\boldsymbol{w}$ is an extreme direction. By adding the feasibility cut to the master problem, the infeasible point $\boldsymbol{x}^*$ is "separated" from the feasible region of the master problem.

The same approach is followed to decompose the RF-MAX formulation (4.31)–(4.36) in Chapter 4 into a master problem and a sub-problem. The only difference compared to the RF-MIN formulation is that the resulting master problem also contains the variables required to facilitate the maximisation of NPV. For the sake of completeness, the resulting master problem for the resource flow RCSP when maximising NPV is presented below. The objective of the *resource flow Benders master problem* (RFM) is to

$$\text{maximise} \sum_{i \in \mathcal{N}} y_i, \tag{6.10}$$

subject to the constraints

$$\begin{aligned}
z_{ij} &= 1, && (i,j) \in \mathcal{Z}, && (6.11)\\
s_j - s_i - (d_i + M)z_{ij} &\geq -M, && (i,j) \in \mathcal{A}, && (6.12)\\
s_i - \sum_{v \in \mathcal{V}} \lambda_{iv} s_{iv} &= 0, && i \in \mathcal{N}, && (6.13)\\
y_i - \sum_{v \in \mathcal{V}} \lambda_{iv} y_{iv} &= 0, && i \in \mathcal{N}, && (6.14)\\
\sum_{v \in \mathcal{V}} \lambda_{iv} &= 1, && i \in \mathcal{N}, && (6.15)\\
\lambda_{i0} - \ell_{i1} &\leq 0, && i \in \mathcal{N}, && (6.16)\\
\lambda_{iv} - \ell_{iv} - \ell_{i(v+1)} &\leq 0, && i \in \mathcal{N}, \quad v \in \mathcal{V} \setminus \{0, V-1\}, && (6.17)\\
\lambda_{i(V-1)} - \ell_{i(V-1)} &\leq 0, && i \in \mathcal{N}. && (6.18)
\end{aligned}$$

6.2.4 The resource flow separation problem (RFSEP)

A separation problem is suggested based on sub-problem (6.8)–(6.9), which produces a Benders cut $(\boldsymbol{b} - A\boldsymbol{x}^*)^T\boldsymbol{w} \geq 0$ if infeasibility is detected in the sub-problem. A slack variable $\alpha \geq 0$ is introduced for this purpose.

The objective of the *resource flow separation problem* (RFSEP) is to

$$\text{minimise } \alpha, \tag{6.19}$$

subject to the constraints

$$\begin{aligned}
\sum_{(i,j)\in\mathcal{A}(i)} f_{ijr} &= v_{ir}/d_i, && i \in \mathcal{N},\ r \in \mathcal{R}, && (6.20)\\
\sum_{(i,j)\in\mathcal{A}(j)} f_{ijr} &= v_{jr}/d_j, && j \in \mathcal{N},\ r \in \mathcal{R}, && (6.21)\\
f_{ijr} + \alpha &\leq \min\{v_{ir}/d_i, v_{jr}/d_j\}z^*_{ij}, && (i,j) \in \mathcal{A},\ r \in \mathcal{R}. && (6.22)
\end{aligned}$$

The vector $\boldsymbol{z}^* \in \{0,1\}^{|\mathcal{A}|}$ is the solution obtained for the linear ordering variables z_{ij} by solving the master problem. It is clear that the value of α is an indicator of feasibility for the sub-problem. If $\alpha > 0$, the sub-problem is infeasible, otherwise it is feasible.

Theorem 6.1. *For a given vector $\boldsymbol{z}^* \in \{0,1\}^{|\mathcal{A}|}$ in the RFSEP, the cut*

$$\sum_{r\in\mathcal{R}}\sum_{(i,j)\in\mathcal{A}} \min\{v_{ir}/d_i, v_{jr}/d_i\}z_{ij}\mu_{ijr} \leq -\sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{N}} v_{ir}/d_i\left(\pi^1_{ir} + \pi^2_{ir}\right), \tag{6.23}$$

where $\boldsymbol{\pi}^1 \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{R}|}$, $\boldsymbol{\pi}^2 \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{R}|}$ and $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{R}|}_{\leq 0}$ are the dual vectors associated with (6.20), (6.21) *and* (6.22) *respectively, is a feasibility cut and separates the infeasible point $\boldsymbol{z}^*$ if $\alpha > 0$.*

Proof. The dual objective function of the RFSEP is to

$$\text{maximise}\left\{\sum_{r\in\mathcal{R}}\left(\sum_{i\in\mathcal{N}}\left(v_{ir}/d_i\pi^1_{ir} + v_{ir}/d_i\pi^2_{ir}\right) + \sum_{(i,j)\in\mathcal{A}} \min\{v_{ir}/d_i, v_{jr}/d_j\}z^*_{ij}\mu_{ijr}\right)\right\}.$$

In the case of infeasibility, $\alpha > 0$ is an optimal solution to the RFSEP and, therefore,

$$\sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{N}} v_{ir}/d_i\left(\pi^1_{ir} + \pi^2_{ir}\right) + \sum_{r\in\mathcal{R}}\sum_{(i,j)\in\mathcal{A}} \min\{v_{ir}/d_i, v_{jr}/d_j\}z^*_{ij}\mu_{ijr} > 0.$$

In order to separate the infeasible point $\boldsymbol{z}^*$, the violated cut

$$\sum_{r\in\mathcal{R}}\sum_{(i,j)\in\mathcal{A}} \min\{v_{ir}/d_i, v_{jr}/d_j\}z_{ij}\mu_{ijr} \leq -\sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{N}} v_{ir}/d_i\left(\pi^1_{ir} + \pi^2_{ir}\right)$$

is added to the master problem. □

6.2.5 Resource flow separation with aggregated constraints (RFSEP-AC)

Recall that the constraints (6.1), which allow the flow variables f_{ijr} to take on values if the corresponding linear ordering variable z_{ij} has been set, may also be written in an aggregated form. For this purpose a separation problem, called the *RFSEP with aggregated constraints* (RFSEP-AC), is introduced.

The objective of the RFSP-AC is to

$$\text{minimise } \alpha, \tag{6.24}$$

subject to the constraints

$$\sum_{(i,j)\in\mathcal{A}(i)} f_{ijr} = v_{ir}/d_i, \qquad i \in \mathcal{N},\ r \in \mathcal{R}, \tag{6.25}$$

$$\sum_{(i,j)\in\mathcal{A}(j)} f_{ijr} = v_{jr}/d_j, \qquad j \in \mathcal{N},\ r \in \mathcal{R}, \tag{6.26}$$

$$\sum_{r\in\mathcal{R}} f_{ijr} + \alpha \leq \left(\sum_{r\in\mathcal{R}} U_r\right) z^*_{ij}, \qquad (i,j) \in \mathcal{A}. \tag{6.27}$$

The vector $\boldsymbol{z}^* \in \{0,1\}^{|\mathcal{A}|}$ is the solution obtained for the linear ordering variables z_{ij} by solving the master problem. If $\alpha > 0$, the sub-problem is infeasible, otherwise it is feasible.

Theorem 6.2. *For a given vector $\boldsymbol{z}^* \in \{0,1\}^{|\mathcal{A}|}$ in the RFSEP-AC, the cut*

$$\sum_{(i,j)\in\mathcal{A}} \left(\sum_{r\in\mathcal{R}} U_r\right) z_{ij}\mu_{ij} \leq -\sum_{i\in\mathcal{N}}\sum_{r\in\mathcal{R}} v_{ri}/d_i(\pi^1_{ir} + \pi^2_{ir}), \tag{6.28}$$

where $\boldsymbol{\pi}^1 \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{R}|}$, $\boldsymbol{\pi}^2 \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{R}|}$ and $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{A}|}_{\leq 0}$ are the dual vectors associated with (6.25), (6.26) *and* (6.27) *respectively, is a feasibility cut and separates the infeasible point $\boldsymbol{z}^*$ if $\alpha > 0$.*

Proof. The dual objective function of RFSEP-AC is to

$$\text{maximise} \left\{ \sum_{i\in\mathcal{N}}\sum_{r\in\mathcal{R}} (v_{ir}/d_i\pi^1_{ir} + v_{ir}/d_i)\pi^2_{ir} + \sum_{(i,j)\in\mathcal{A}} \left(\sum_{r\in\mathcal{R}} U_r\right) z^*_{ij}\mu_{ij} \right\}. \tag{6.29}$$

In the case of infeasibility, $\alpha > 0$ is an optimal solution to the RFSEP-AC and, therefore,

$$\sum_{i\in\mathcal{N}}\sum_{r\in\mathcal{R}} v_{ir}/d_i(\pi^1_{ir} + \pi^2_{ir}) + \sum_{(i,j)\in\mathcal{A}} \left(\sum_{r\in\mathcal{R}} U_r\right) z^*_{ij}\mu_{ij} > 0. \tag{6.30}$$

In order to separate the infeasible point $\boldsymbol{z}^*$, the violated cut

$$\sum_{(i,j)\in\mathcal{A}} \left(\sum_{r\in\mathcal{R}} U_r\right) z_{ij}\mu_{ij} \leq -\sum_{i\in\mathcal{N}}\sum_{r\in\mathcal{R}} v_{ir}/d_i(\pi^1_{ir} + \pi^2_{ir}) \tag{6.31}$$

is added to the master problem. □

6.2.6 The disaggregated resource flow separation problem (RFSEP-DA)

The constraints in RFSEP are separable with the result that the flow separation problem may be decomposed into independent sub-problems. Specifically, for each $r \in \mathcal{R}$, the objective of the *disaggregated resource flow separation problem* (RFSEP-DA), is to

$$\text{minimise } \alpha_r, \tag{6.32}$$

subject to the constraints

$$\sum_{(i,j)\in\mathcal{A}(i)} f_{ijr} = v_{ir}/d_i, \qquad i \in \mathcal{N}, \tag{6.33}$$

$$\sum_{(i,j)\in\mathcal{A}(j)} f_{ijr} = v_{jr}/d_j, \qquad j \in \mathcal{N}, \tag{6.34}$$

$$f_{ijr} + \alpha_r \leq \min\{v_{ir}/d_i, v_{jr}/d_j\}z^*_{ij}, \qquad (i,j) \in \mathcal{A}. \tag{6.35}$$

If $\alpha_r > 0$ for any $r \in \mathcal{R}$, then the current vector $\boldsymbol{z}^*$ is infeasible.

Theorem 6.3. *For a given $r \in \mathcal{R}$ and a vector $\boldsymbol{z}^* \in \{0,1\}^{|\mathcal{A}|}$, the cut*

$$\sum_{(i,j)\in\mathcal{A}} \min\{v_{ir}/d_i, v_{jr}/d_j\}z_{ij}\mu_{ij} \leq -\sum_{i\in\mathcal{N}} v_{ir}/d_i \left(\pi_i^1 + \pi_i^2\right), \tag{6.36}$$

where $\boldsymbol{\pi}^1 \in \mathbb{R}^{|\mathcal{N}|}$, $\boldsymbol{\pi}^2 \in \mathbb{R}^{|\mathcal{N}|}$ and $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{A}|}_{\leq 0}$ are the dual vectors associated with (6.33), (6.34) *and* (6.35) *respectively, separates the infeasible point $\boldsymbol{z}^*$ if $\alpha_r > 0$.*

Proof. The dual objective function of the RFSEP-DA for a given $r \in \mathcal{R}$, is to

$$\text{maximise} \left\{ \sum_{i\in\mathcal{N}} \left(v_{ir}/d_i\pi_i^1 + v_{ir}/d_i\pi_i^2\right) + \sum_{(i,j)\in\mathcal{A}} \min\{v_{ir}/d_i, v_{jr}/d_j\}z^*_{ij}\mu_{ij} \right\}.$$

In the case of infeasibility, $\alpha > 0$ is an optimal solution to the RFSEP-DA and, therefore,

$$\sum_{i\in\mathcal{N}} v_{ir}/d_i \left(\pi_i^1 + \pi_i^2\right) + \sum_{(i,j)\in\mathcal{A}} \min\{v_{ir}/d_i, v_{jr}/d_j\}z^*_{ij}\mu_{ij} > 0.$$

In order to separate the infeasible point $\boldsymbol{z}^*$, the violated cut

$$\sum_{(i,j)\in\mathcal{A}} \min\{v_{ir}/d_i, v_{jr}/d_j\}z_{ij}\mu_{ij} \leq -\sum_{i\in\mathcal{N}} v_{ir}/d_i \left(\pi_i^1 + \pi_i^2\right)$$

is added to the master problem. □

6.2.7 The single-LP disaggregated separation problem (RFSEP-SLP)

An extension of the disaggregated resource flow separation problem presented above, is the *single-LP disaggregated resource flow separation problem* (RFSEP-SLP). Conceptually, the implementation of the feasibility cuts (6.36) may be achieved through the solution of a single LP model in an iterative fashion. More specifically, instead of having $|\mathcal{R}|$ LP models in computer memory, the same LP problem is solved but each time for a different resource $r \in \mathcal{R}$. The right-hand-side values of constraints (6.33) and (6.34) are modified each time the LP is solved for a different resource.

The anticipated advantage of a single-LP implementation is the efficient use of memory. The disadvantage, on the other hand, is that valuable computing time is spent updating the right-hand-side values of constraints (6.33) and (6.34) each time a different resource is considered.

6.2.8 The parallel disaggregated separation problem (RFSEP-PDA)

Another extension of the disaggregated resource flow separation problem, is the *parallel disaggregated resource flow separation problem* (RFSEP-PDA). A parallel implementation of the feasibility cuts (6.36) is possible by allowing the separation problem (6.32)–(6.35) to be solved in parallel for each resource $r \in \mathcal{R}$. Computational benefits are expected when solving the RCSP on a multi-processing platform and, especially, for problem instances with a large number of resources.

6.2.9 Explicit cutset inequalities (ECS)

Valid inequalities have been applied successfully to many large-scale MILP problems for improving computing times (see, for example, Cornuéjols [26]). In the context of a branch-and-cut approach, valid inequalities may either be added explicitly to the master problem, or implicitly through the use of separation routines.

Of specific interest to this study is the use of so-called *network cutset inequalities* (see, for example, [8, 13, 37]). Let $\mathcal{V} \subseteq \mathcal{N}$ be a subset of activities and let $\bar{\mathcal{V}} = \mathcal{N} \setminus \mathcal{V}$ be its complement. The set of arcs $\mathcal{A}(\mathcal{V}, \bar{\mathcal{V}})$, where each arc is incident from an activity in $\mathcal{V}$ and incident to an activity in $\bar{\mathcal{V}}$, is called a *cut*.

For each arc $(i, j) \in \mathcal{A}$, there exists a linear ordering variable z_{ij}. Recall from the resource flow formulation above that the use of the variable z_{ij} is to allow the corresponding flow variables f_{ijr} to take on values, if $z_{ij} = 1$. Conceptually, the variable z_{ij} may be interpreted as a capacity variable within a network flow context.

Proposition 6.1. *For a specific resource $r \in \mathcal{R}$ and a subset $\mathcal{V}$, the valid inequality*

$$\sum_{(i,j)\in\mathcal{A}(\mathcal{V},\bar{\mathcal{V}})} \min\{v_{ir}/d_i, v_{jr}/d_j\} z_{ij} \;\geq\; \max_{i\in\mathcal{V}}\{v_{ir}/d_i\} \tag{6.37}$$

must hold in order for any given solution $\boldsymbol{z}^*$ to be feasible.

Proof. Consider the trivial case where $\mathcal{V}$ contains only the activity $i \in \mathcal{N}$. According to constraints (4.8) in the RF-MIN problem formultion of Section 4.2.2, the sum of the resource flows emanating from activity i should match the resource requirement of i. More specifically,

$$\sum_{(i,j)\in\mathcal{A}(i)} f_{ijr} \;=\; v_{ir}/d_i. \tag{6.38}$$

Substituting f_{ijr} by $\min\{v_{ir}/d_i, v_{jr}/d_j\}z_{ij}$ according to constraints (4.10), and allowing the resource requirement v_{ir}/d_i to be written as $\max_{i\in\mathcal{V}}\{v_{ir}/d_i\}$ since $\mathcal{V}$ only contains the one activity i, the equality (6.38) may be written as

$$\sum_{(i,j)\in\mathcal{A}(i)} \min\{v_{ir}/d_i, v_{jr}/d_j\} z_{ij} \;\geq\; \max_{i\in\mathcal{V}}\{v_{ir}/d_i\}. \tag{6.39}$$

Valid inequality (6.39) is equivalent to (6.37) since the arcs $(i, j) \in \mathcal{A}(\mathcal{V}, \bar{\mathcal{V}})$ correspond to the arcs emanating from activity i.

The right-hand term $\max_{i\in\mathcal{V}}\{v_{ir}/d_i\}$ in (6.37) is justified for the case when $|\mathcal{V}| > 1$. Consider two activities $i \in \mathcal{V}$ and $j \in \mathcal{V}$ that are adjacent according to the resource flow graph $G(\mathcal{N}, \mathcal{A})$. For any solution with $f_{ijr} > 0$ or $f_{jir} > 0$, the total flow from $\mathcal{V}$ is less than $v_{ir} + v_{jr}$. □

The valid inequalities (6.37) are referred to as *explicit cutset* (ECS) inequalities, since the resource requirements are explicitly considered in the formulation of these inequalities. An implicit version of these inequalities is presented in the next section. For the purpose of evaluating computational efficiency, the ECS inequalities are generated up-front and added explicitly to the Benders master problem RFM. Due to the large number of different cut combinations, ECS inequalities are generated only for the case where $|\mathcal{V}| = 1$.

6.2.10 Implicit cutset inequalities (ICS)

The anticipated disadvantage of employing the explicit cutset inequalities (6.37) is that they are derived for each resource $r \in \mathcal{R}$. For a large number of resources, this may result in an inflated Benders master problem which may, in turn, have a deteriorating effect on computing times. An alternative cutset derivation is provided that takes resource requirements into account implicitly.

Proposition 6.2. *For any subset $\mathcal{V}$ the valid inequality*

$$\sum_{(i,j)\in\mathcal{A}(\mathcal{V},\bar{\mathcal{V}})} z_{ij} \geq |\mathcal{V}| \tag{6.40}$$

must hold in order for any given solution $\boldsymbol{z}^*$ to be feasible.

Proof. The trivial case is where each activity $\mathcal{V}$ is the predecessor of one or more activities in $\bar{\mathcal{V}}$ according to the precedence graph $G(\mathcal{N},\mathcal{Z})$. For the case where there is only one arc $(i,j) \in \mathcal{A}(\mathcal{V},\bar{\mathcal{V}})$ which forces precedence, that is $(i,j) \in \mathcal{Z}$, precedence for the remaining arcs $\mathcal{A}(\mathcal{V},\bar{\mathcal{V}}) \setminus (i,j)$ are implied since the activities $\bar{\mathcal{V}} \setminus \{j\}$ are directly or indirectly preceded by the activity $j \in \bar{\mathcal{V}}$. □

Instead of adding the cutset inequalities (6.40) explicitly to the resource flow formulation, a separation routine is employed which identifies a minimum cut by solving a maximum flow problem. Let z^*_{ij} be the LP relaxation solution for the resource flow RCSP at a branch-and-bound node, for each $(i,j) \in \mathcal{A}$. By employing the preflow–push algorithm by Goldberg and Tarjan [40] with z^*_{ij} as the arc capacities, a minimum cut $\mathcal{A}(\mathcal{V},\bar{\mathcal{V}})$ is obtained and used to derive the valid inequality (6.40).

6.2.11 The right-shift primal heuristic (RSPH)

The Benders decomposition solution approach outlined above involves the resource flow master problem RFM and the application of one of the resource flow separation routines, RFSEP, RFSEP-AC, RFSEP-DA, RFSEP-SLP or RFSEP-PDA. The purpose of the resource flow separation routines is to ensure feasibility. The branch-and-cut approach, outlined in Figure 6.1, suggests the use of a heuristic for generating primal solutions on completion of the separation process.

The proposed *right-shift primal heuristic* (RSPH) is closely related to the TRSH introduced in Section 4.5.1. The main difference is that it relies on current information obtained from the LP relaxation of the Benders master problem, instead of generating feasible solutions from scratch. Consider the RFM formulation (6.10)–(6.18). During the solution of the RFM at a node of the branch-and-bound tree, solution values are obtained for the start time variables $\boldsymbol{s}$, which may not necessarily constitute a feasible solution. That is, some of the linear ordering variables $\boldsymbol{z}$ may be fractional. The motivation for the suggested RSPH is that the LP relaxation solutions

Algorithm 6.1: The right-shift primal heuristic (RSPH)

```
N^B = ∅.
N^P = {0}.
s*_0 = 0.
for (0, j) ∈ Z do
  N^B = N^B ∪ {j}.
end for
t = 0
while N^B <> ∅ do
  period_feasible = false.
  for i ∈ N^B do
    if φ(i,t,t)v_ir ≤ U_r − Σ_{i∈N^P} φ(i,t,s*_i)v_ir  ∀r ∈ R  then
      period_feasible = true.
    end if
  end for
  if period_feasible = true then
    Sort N^B in ascending order according to s*_LP.
    for i ∈ N^B do
      if φ(i,t,t)v_ir ≤ U_r − Σ_{i∈N^P} φ(i,t,s*_i)v_ir  ∀r ∈ R  then
        if s*_j + d_j ≤ S^T_t  ∀(i,j) ∈ Z  then
          s*_i = S^T_t
          N^P = N^P ∪ {i}
          N^B = N^B \ {i}
          for (i,j) ∈ Z do
            N^B = N^B ∪ {j}.
          end for
        end if
      end if
    end for
  end if
  t=t+1
end while
for (i,j) ∈ A do
  if s*_j ≥ s*_i + d_i then
    z*_ij = 1
  end if
end for
Solve RFM(z*).
```

of the start time variables, denoted by $\boldsymbol{s}^*_{LP}$, may give a good indication of an optimal solution, since they are directly influenced by the objective function. More specifically, the sequencing order of the activities implied by the start time solutions $\boldsymbol{s}^*_{LP}$ may possibly correspond to the same order as in an optimal solution. The approach of the RSPH is, therefore, to right-shift activities by maintaining the sequencing order implied by the start time solutions.

Keeping with the notation introduced in Section 4.5.1, the set $\mathcal{N}^B$ contains activities that are eligible to start in the current time period $t \in \mathcal{T}$ and the set $\mathcal{N}^P$ contains activities already processed and for which start time values s^*_i have been determined. Once an activity i has been selected for execution at a time period t, its start time is set to $s^*_i = S^{\mathcal{T}}_t$, it is added to $\mathcal{N}^P$ and it is removed from $\mathcal{N}^B$. An activity j may only be considered eligible to start during a time period t if all of its predecessors i have been completed prior to t. More specifically, $\mathcal{N}^B = \mathcal{N}^B \cup \{j\}$ only if $i \in \mathcal{N}^P$ for all $(i,j) \in \mathcal{Z}$, and only if $s_j \geq s_i + d_i$.

Selecting activities from the eligibility set $\mathcal{N}^B$ for execution during time period t is based on the activity ranking provided by the LP relaxation solution $\boldsymbol{s}^*_{LP}$. That is, the activity with the smallest start time LP solution is selected first. Activities are selected from the eligibility set $\mathcal{N}^B$ until the resource capacity for the current time period t has been depleted.

It is important to note that the start time solutions $\boldsymbol{s}^*$ generated by the RSPH heuristic are not the final solutions utilised by the branch-and-cut process. The solution vector $\boldsymbol{s}^*$ is only used to derive the proposed sequencing order of a potential feasible solution. More specifically, the solution $z^*_{ij} = 1$ is constructed for all resource flow arcs $(i,j) \in \mathcal{A}$, provided that $s^*_j \geq s^*_i + d_i$. The solution vector $\boldsymbol{z}^*$ is then used to fix the ordering variables $\boldsymbol{z}$ in a copy of the RFM problem. The notation RFM($\boldsymbol{z}^*$) is used to indicate the solution of the RFM problem with the ordering variables $\boldsymbol{z}$ fixed to the values provided by the vector $\boldsymbol{z}^*$. The solution obtained by solving RFM($\boldsymbol{z}^*$) is used as a primal solution of the Benders master problem RFM within the branch-and-cut framework. The complete algorithm for the RSPH is provided in pseudocode form in Algorithm 6.1.

6.3 Computational results

The empirical study for this chapter involves the data sets described in Section 5.2 (having extended activity durations), as well as newly created data sets. For the purpose of testing the scalability of the newly proposed formulations and algorithms, additional data sets were created to conform to typical mining problem instances with a large number of resources. In all of the problem instances considered up to now, a maximum number of resources of five was considered. The J30, J60 and J90 data sets all have four resources per problem instance, where as the BL and CN data sets have on average 3 and 2.85 resources per problem instance, respectively.

Creating problem instances with more resources required the generation of new resource upper limits U_r and resource requirements v_{ir}, based on existing problem instances. For example, a new resource $\tilde{r} \in \mathcal{R}$ was created from an existing resource $r \in \mathcal{R}$, by letting $U_{\tilde{r}} = U_r(1+u)$ with $u \sim U(-0.5, 0.5)$ and by letting $v_{i\tilde{r}} = v_{ir}(1+n_i)$ with $n_i \sim N(0,1)$.

The newly created data sets with randomly generated resources are called the J30(50)DR×10, J60(50)DR×10, J90(50)DR×10, CNDR×10 and BLDR×10 data sets. The post-fix R×10 indicates that the resulting number of resources for each of the problem instances are ten times more than the number of resources in the original problem instances. For example, the problem instances of the J30(50)D data set have a total of 4 resources each, whereas the J30(50)DR×10 data set have a total of 40 resources per problem instance.

Data set	number of activities	avg. activity duration	*OS* avg	*OS* σ	*NC* avg	*NC* σ	*RF* avg	*RF* σ	*RS* avg	*RS* σ	*DR* avg	*DR* σ
J30(50)DR×10	30	252.7	0.45	0.07	1.64	0.21	0.3	0.14	0.44	0.19	0.78	0.15
J60(50)DR×10	60	257.8	0.31	0.04	1.58	0.13	0.33	0.16	0.19	0.18	0.72	0.18
J90(50)DR×10	90	264.7	0.26	0.03	1.56	0.12	0.35	0.15	0.11	0.11	0.67	0.19
BLDR×10	22-27	140.0	0.34	0.07	2.93	0.46	0.3	0.03	0.47	0.09	0.69	0.08
CNDR×10	17-35	309.3	0.23	0.07	1.8	0.27	0.48	0.03	0.24	0.10	0.91	0.04

TABLE 6.1: *Tractability indicators for the randomly generated problem instances with an increase in the number of resources by a factor of ten.*

Statistics on the indicators for the newly created data sets are listed in Table 6.1. The most significant difference, when compared to the statistics of the original data sets provided in Table 5.1, is that the average DR values are now much higher. Although this may suggest that the problem instances are now "easier," increased computing times are expected due to the increased number of variables and constraints required in the RF formulation in order to accommodate the higher number of resources.

The computational results reported in all subsequent tables are first ranked according to the percentage of instances for which optimal solutions could be computed, then according to the average computing time it took to compute the optimal solutions, the percentage of instances for which feasible solutions could be computed, and finally, according to the average integrality gap obtained for all feasible solutions computed.

The results to follow are only for the objective of maximising NPV. This is motivated by the importance of using NPV as a performance measure in the mining industry, for evaluating competing scheduling solutions.

6.3.1 Variable and constraint reduction

The results obtained by applying the variable and constraint reduction approaches described in Section 6.1, are presented in Tables 6.2 and 6.3. The results in Table 6.2 are for the newly created data sets having extended activity durations, that is, the data sets with a post-fix "D."

The abbreviation RF-AC refers to the aggregation of constraints in the RF formulation according to constraint (6.2). The abbreviation RF-GR refers to the use of the reduced resource flow graph $G'(\mathcal{N}, \mathcal{A}')$ in the RF formulation. The abbreviation RF-AC-GR is used to indicate the combination of the two approaches. The different options of combining the constraint aggregation and graph reduction approaches are also considered in conjunction with the *time-indexed right shift heuristic* (TRSH) described in Section 4.5.1. Note that since the maximisation of NPV is considered as the objective function of choice in the RF formulation in the remainder of this dissertation, only the feasible solutions obtained from the TRSH are usable. The upper bound T^* on the scheduling horizon T obtained from the TRSH may be invalid due to the maximisation of NPV (see the discussion in Section 4.5.1).

The results in Table 6.2 clearly show the computational benefit of applying the variable and constraint reduction approaches described in Section 6.1. Improvements in computing times are apparent for all of the data sets where constraint aggregation or graph reduction or both approaches were applied. The added advantage of using the TRSH is also clear and in most cases it produced feasible solutions for all of the problem instances.

The results computed for the newly created data sets with a large number of resources are provided in Table 6.3. As a first observation, computing times are noticeably worse due to an increase in the number of resources. For example, the percentage of instances solved to optimality decreased from 74% to 54% when comparing the results for the J30(50)D data set with those of the J30(50)DR×10 data set. Similar reductions in the percentage optimal solutions were recorded for the other data sets. Although the plain RF formulation led to the exact solution of more problem instances of the J30(50)DR×10 data set, the average gap for computing feasible solutions decreased from 19.1% to 9.2% when employing graph reduction in conjunction with the TRSH. Improvements were also observed for the J60(50)DR×10 data set, where the combination of graph reduction and TRSH resulted in feasible solutions for all of the problem instances, compared to the plain RF formulation which yielded feasible solutions to only 74% of the problem instances. The last panel of Table 6.3 provides results compiled over all of the

Data set	Model	Heuristic	Solved to optimality Instances (%)	Solved to optimality Time (s)	Feasible Solutions Instances (%)	Feasible Solutions Gap (%)
J30(50)D	RF-AC-GR	TRSH	74	40.3	98	0.5
	RF-GR	TRSH	74	44.3	96	0.4
	RF-AC	—	72	29.8	100	0.5
	RF-AC	TRSH	72	36.8	98	0.5
	RF-AC-GR	—	70	31.0	100	0.6
	RF	—	68	44.5	92	0.5
	RF-GR	—	68	47.0	92	0.5
J60(50)D	RF-GR	TRSH	34	151.9	100	7.6
	RF-AC-GR	—	30	106.4	100	10.3
	RF-AC	—	30	112.3	100	9.9
	RF-AC-GR	TRSH	30	115.5	100	7.5
	RF-GR	—	28	105.3	98	8.3
	RF-AC	TRSH	26	58.5	100	7.7
	RF	—	26	84.3	98	8.8
J90(50)D	RF-GR	TRSH	8	396.6	100	9.9
	RF-GR	—	6	78.2	74	17.0
	RF-AC	TRSH	6	130.7	100	10.1
	RF-AC	—	6	274.2	98	16.8
	RF-AC-GR	—	6	420.2	98	17.4
	RF	—	4	64.8	72	17.4
	RF-AC-GR	TRSH	4	91.5	100	10.1
BLD	RF-GR	TRSH	92.3	7.6	100	0.1
	RF-GR	—	92.3	8.7	100	0.1
	RF	—	92.3	8.9	100	0.1
	RF-AC	—	89.7	6.2	97.4	0.1
	RF-AC-GR	TRSH	89.7	7.1	97.4	0.1
	RF-AC	TRSH	87.2	7.1	94.9	0.1
	RF-AC-GR	—	87.2	8.2	94.9	0.1
CND	RF-GR	—	16.1	55.8	98.2	7.0
	RF-AC	TRSH	16.1	65.3	98.2	6.6
	RF-AC-GR	TRSH	16.1	71.4	100	6.8
	RF-AC-GR	—	16.1	82.5	100	7.4
	RF-GR	TRSH	14.3	44.0	98.2	6.5
	RF	—	14.3	59.0	98.2	6.7
	RF-AC	—	14.3	67.6	92.9	7.8

TABLE 6.2: *The effectiveness of applying the variable and constraint reduction approaches to the RF formulation.*

data sets considered. In summary, the use of the graph reduction and constraint aggregation approaches resulted in the largest percentage of problem instances solved to optimality, whereas the combined use of the TRSH resulted in the largest percentage of problem instances for which at least one feasible solution could be computed.

6.3.2 Benders decomposition

The proposed Benders decomposition approach, as outlined earlier, entails solving a master problem and an accompanying sub-problem. The sub-problem, which is used for generating the necessary feasibility cuts, involves only the resource flow-related variables and constraints. The anticipated advantage is that scalability may be improved due to the large number of flow variables that are projected out of the master problem.

The results for the Benders decomposition of the RCSP are presented in two parts. First, computational results are presented for the purpose of evaluating the computational efficiency

			Solved to optimality		Feasible Solutions	
Data set	Model	Heuristic	Instances (%)	Time (s)	Instances (%)	Gap (%)
J30(50)DR×10	RF	—	54	47.5	100	19.1
	RF-GR	TRSH	52	34.4	100	9.2
	RF-AC-GR	TRSH	52	43.7	100	7.9
	RF-AC	—	50	30.0	98	5.7
	RF-GR	—	50	31.1	96	19.7
	RF-AC	TRSH	50	32.4	96	6.5
	RF-AC-GR	—	46	23.7	94	8.2
J60(50)DR×10	RF-GR	TRSH	20	135.0	100	33.3
	RF	—	16	139.2	74	59.6
	RF-AC-GR	TRSH	16	163.6	98	32.2
	RF-AC	TRSH	16	182.9	98	33.1
	RF-AC	—	14	104.5	94	59.2
	RF-GR	—	14	105.4	88	59.5
	RF-AC-GR	—	14	162.3	94	56.8
J90(50)DR×10	RF-AC-GR	—	2	348.5	54	48.4
	RF-AC	TRSH	0	—	98	28.7
	RF-GR	TRSH	0	—	96	29.7
	RF-AC-GR	TRSH	0	—	94	28.2
	RF-AC	—	0	—	50	40.4
	RF-GR	—	0	—	40	47.4
	RF	—	0	—	26	34.2
BLDR×10	RF-GR	—	76.9	46.1	100	1.0
	RF	—	76.9	50.5	100	0.8
	RF-AC	TRSH	76.9	61.4	94.9	1.0
	RF-GR	TRSH	74.4	51.8	92.3	0.8
	RF-AC-GR	TRSH	71.8	58.4	94.9	1.0
	RF-AC-GR	—	69.2	58.0	87.2	0.7
	RF-AC	—	66.7	53.8	94.9	1.2
CNDR×10	RF-AC	TRSH	3.6	100.3	98.2	36.9
	RF-GR	—	3.6	121.7	100	105.8
	RF-AC	—	3.6	121.9	96.4	80.0
	RF	—	3.6	147.9	100	855.3
	RF-GR	TRSH	3.6	152.3	100	48.3
	RF-AC-GR	—	3.6	172.6	100	137.0
	RF-AC-GR	TRSH	1.8	39.5	98.2	40.1
All	RF-GR	—	20.8	28.6	84.5	50.6
	RF-AC	—	20	35.1	86.5	39.8
	RF	—	18.4	20.7	79.2	265.6
	RF-AC-GR	TRSH	18.4	26.1	97.1	23.3
	RF-GR	TRSH	18	20.5	98	26.2
	RF-AC-GR	—	17.1	23.8	86.1	57.2
	RF-AC	TRSH	16.7	29.0	97.1	22.7

TABLE 6.3: *The effectiveness of applying the variable and constraint reduction approaches when considering the newly created problem instances with a larger number of resources.*

of the proposed separation routines. This is followed by results demonstrating the effectiveness of the proposed valid inequalities and the proposed primal heuristic. The positive results obtained previously by means of the TRSH for generating initial feasible solutions justify the use of the heuristic in terms of providing starting feasible solutions to the Benders master problem. All of the computational results presented in this section, therefore, include the use of the TRSH.

Table 6.4 provides computational results for applying the various separation routines as part of Benders decomposition. The problem instances were solved by considering the four different separation sub-problems, namely the RFSEP, RFSEP-AC, RFSEP-DA, RFSEP-SLP and

Data set	Model	Separation	Solved to optimality Instances (%)	Time (s)	Feasible Solutions Instances (%)	Gap (%)
J30(50)D	RFM-GR	RFSEP	40	25.3	100	193.9
	RFM	RFSEP	40	29.6	100	154.8
	RFM-GR	RFSEP-PDA	38	18.2	100	243.7
	RFM	RFSEP-PDA	38	21.3	100	261.1
	RFM	RFSEP-DA	38	21.7	100	261.1
	RFM-GR	RFSEP-SLP	36	8.2	100	164.5
	RFM-GR	RFSEP-DA	36	8.4	100	265.7
	RFM	RFSEP-SLP	36	11.4	100	262.7
	RFM-GR	RFSEP-AC	36	18.5	100	182.7
	RFM	RFSEP-AC	36	31.1	100	166.5
J60(50)D	RFM	RFSEP	4	241.3	100	108.7
	RFM-GR	RFSEP	2	123.6	100	108.7
	RFM	RFSEP-PDA	2	145.8	100	112.7
	RFM	RFSEP-DA	2	165.3	100	113.2
	RFM-GR	RFSEP-PDA	2	217.8	102	112.5
	RFM-GR	RFSEP-AC	2	299.1	100	117.3
	RFM	RFSEP-SLP	2	444.0	100	114.4
	RFM	RFSEP-AC	2	472.6	100	117.4
	RFM-GR	RFSEP-DA	2	512.3	100	116.4
	RFM-GR	RFSEP-SLP	0	—	100	114.1
J90(50)D	RFM	RFSEP-SLP	0	—	100	90.3
	RFM-GR	RFSEP-SLP	0	—	100	90.3
	RFM	RFSEP-DA	0	—	100	90.4
	RFM	RFSEP-PDA	0	—	100	90.4
	RFM-GR	RFSEP	0	—	100	90.5
	RFM	RFSEP-AC	0	—	100	90.5
	RFM-GR	RFSEP-DA	0	—	100	90.5
	RFM	RFSEP	0	—	100	90.6
	RFM-GR	RFSEP-PDA	0	—	100	90.6
	RFM-GR	RFSEP-AC	0	—	100	90.7
BLD	RFM	RFSEP	61.5	55.5	100	2.4
	RFM-GR	RFSEP	59	50.4	100	2.6
	RFM	RFSEP-PDA	48.7	89.0	100	4.8
	RFM	RFSEP-DA	48.7	90.8	100	4.6
	RFM-GR	RFSEP-PDA	46.2	76.0	100	4.3
	RFM-GR	RFSEP-DA	43.6	60.5	100	4.4
	RFM	RFSEP-SLP	46.2	77.2	100	13.3
	RFM-GR	RFSEP-SLP	43.6	81.5	100	5
	RFM	RFSEP-AC	24	73.9	78	20.3
	RFM-GR	RFSEP-AC	22	81.2	78	20.7
CND	RFM	RFSEP	7.1	169.9	100	133.9
	RFM-GR	RFSEP	5.4	135.3	100	124.5
	RFM	RFSEP-SLP	3.6	149.9	100	143.1
	RFM-GR	RFSEP-SLP	3.6	158.3	100	145.2
	RFM	RFSEP-AC	3.6	89.4	100	137.6
	RFM-GR	RFSEP-PDA	3.6	117.3	100	138.2
	RFM	RFSEP-PDA	3.6	120.7	100	134.4
	RFM	RFSEP-DA	3.6	125.9	100	134.3
	RFM-GR	RFSEP-DA	4	129.9	112	170.7
	RFM-GR	RFSEP-AC	4	146.4	112	141.9

TABLE 6.4: *Comparison of different separation routines for generating Benders feasibility cuts.*

			Solved to optimality		Feasible Solutions	
Data set	Model	Separation	Instances (%)	Time (s)	Instances (%)	Gap (%)
J30(50)DR×10	RFM-GR	RFSEP-PDA	30	3.4	98	133.7
	RFM-GR	RFSEP-DA	30	4.3	98	130.2
	RFM	RFSEP-PDA	30	4.5	98	178.5
	RFM-GR	RFSEP-SLP	30	5.0	98	176.6
	RFM	RFSEP-SLP	30	5.1	98	124.1
	RFM	RFSEP-DA	30	5.3	98	123.6
	RFM-GR	RFSEP-AC	30	29.7	98	119.9
	RFM-GR	RFSEP	30	30.8	98	117.5
	RFM	RFSEP-AC	30	34.9	98	161.2
	RFM	RFSEP	30	38.9	98	177.8
J60(50)DR×10	RFM-GR	RFSEP-PDA	0	—	200	87.5
	RFM	RFSEP-PDA	0	—	200	89.8
	RFM-GR	RFSEP-DA	0	—	100	88.3
	RFM	RFSEP-SLP	0	—	100	90.9
	RFM-GR	RFSEP-SLP	0	—	100	91.2
	RFM	RFSEP-DA	0	—	100	91.4
	RFM-GR	RFSEP	0	—	70	91.3
	RFM	RFSEP	0	—	70	92.5
	RFM-GR	RFSEP-AC	0	—	46	93.8
	RFM	RFSEP-AC	0	—	46	94
J90(50)DR×10	RFM	RFSEP-PDA	0	—	100	79.5
	RFM	RFSEP-DA	0	—	100	79.7
	RFM-GR	RFSEP-DA	0	—	100	79.8
	RFM-GR	RFSEP-PDA	0	—	100	80
	RFM-GR	RFSEP-SLP	0	—	100	80.1
	RFM	RFSEP-SLP	0	—	100	80.2
	RFM-GR	RFSEP	0	—	32	88.6
	RFM	RFSEP	0	—	22	85.3
	RFM-GR	RFSEP-AC	0	—	22	85.9
	RFM	RFSEP-AC	0	—	22	86.1
BLDR×10	RFM-GR	RFSEP	38.5	79.9	100	60.7
	RFM	RFSEP	35.9	89.4	100	97.7
	RFM-GR	RFSEP-DA	30.8	66.7	100	231.4
	RFM-GR	RFSEP-PDA	28.2	36.5	100	575.9
	RFM	RFSEP-PDA	28.2	44.9	100	195.7
	RFM	RFSEP-DA	28.2	48.5	100	340.7
	RFM-GR	RFSEP-SLP	30.8	59.5	100	82.1
	RFM	RFSEP-SLP	28.2	63.3	100	1320.5
	RFM-GR	RFSEP-AC	12	137.0	78	2068.9
	RFM	RFSEP-AC	10	175.7	78	558.5
CNDR×10	RFM	RFSEP-SLP	0	—	100	145.8
	RFM-GR	RFSEP-SLP	0	—	100	165.4
	RFM	RFSEP-PDA	0	—	100	145.4
	RFM	RFSEP-DA	0	—	100	145.9
	RFM-GR	RFSEP-DA	0	—	100	146
	RFM	RFSEP	0	—	100	149.6
	RFM-GR	RFSEP	0	—	100	150
	RFM-GR	RFSEP-PDA	0	—	100	150.1
	RFM-GR	RFSEP-AC	0	—	102	207.7
	RFM	RFSEP-AC	0	—	98	217.4

TABLE 6.5: *Comparison of different separation routines for generating Benders feasibility cuts when considering the newly created problem instances with a larger number of resources.*

			Solved to optimality		Feasible Solutions	
Data set	Inequalities	Heuristic	Instances (%)	Time (s)	Instances (%)	Gap (%)
J30(50)D	ECI-ICI	RSPH	52	31.5	100	1.7
	ECI-ICI	—	52	31.6	100	10.5
	ICI	RSPH	52	34.8	100	1.8
	ECI	RSPH	52	38.1	100	1.6
	ECI	—	50	23.5	100	10.2
	—	RSPH	50	36.1	100	1.9
	ICI	—	48	26.2	100	7.3
J60(50)D	ECI	RSPH	18	179.2	100	1.4
	—	RSPH	18	195.8	100	1.5
	ECI-ICI	—	16	259.5	100	18
	ICI	RSPH	14	154.6	100	1.5
	ECI-ICI	RSPH	14	160.9	100	1.4
	ECI	—	14	183.5	100	18
	ICI	—	12	256.6	100	18.1
J90(50)D	ECI	RSPH	4	169.2	100	1.3
	—	RSPH	4	372.4	100	1.3
	ICI	RSPH	4	392.9	100	1.4
	ECI-ICI	RSPH	2	188.5	100	1.3
	ECI-ICI	—	2	262.7	100	13.3
	ECI	—	0	—	100	13.3
	ICI	—	0	—	100	13.6
BLD	ICI	—	84.6	45.1	100	0.2
	ECI	—	84.6	45.3	100	0.2
	ECI-ICI	—	84.6	45.3	100	0.2
	—	RSPH	82.1	51.8	100	0.2
	ECI	RSPH	82.1	51.8	100	0.2
	ICI	RSPH	82.1	52.0	100	0.2
	ECI-ICI	RSPH	82.1	52.0	100	0.2
CND	ICI	—	10.7	152.4	100	63.3
	ICI	RSPH	10.7	159.2	100	14.9
	ECI-ICI	—	8.9	113.4	100	63.2
	ECI-ICI	RSPH	8.9	140.4	100	16.6
	—	RSPH	8.9	164.8	100	17.6
	ECI	—	7.1	58.2	100	66.8
	ECI	RSPH	7.1	82.9	100	18

TABLE 6.6: *The computational efficiency of cutset inequalities and primal heuristics applied within the branch-and-cut framework.*

RFSEP-PDA. The abbreviation RFM-GR is used to indicate the use of graph reduction in the formulation of the Benders master problem RFM. The use of graph reduction in the RFM also implies a reduction in the number of variables and constraints of the separation sub-problem.

The most notable result from Table 6.4 is that far fewer problem instances were solved to optimality with Benders decomposition, compared to using CPLEX for solving the RF model as a stand-alone problem (see Table 6.2). Although this may appear discouraging, results to follow are more promising in terms of the Benders decomposition approach. Despite the inability of Benders decomposition to produce optimal solutions, it is able to compute feasible solutions for all of the problem instances.

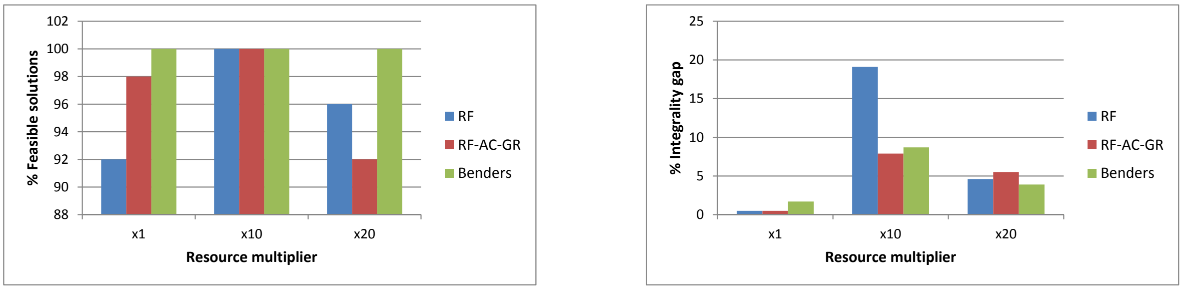
Another apparent result from Table 6.4 is that the RFSEP separation routine dominated for most of the data sets that were considered. The J90(50)D data set is the only data set for which the RFSEP-SLP appears to be efficient. There is, however, no clear indication of the benefit in applying graph reduction.

Data set	Inequalities	Heuristic	Solved to optimality Instances (%)	Solved to optimality Time (s)	Feasible Solutions Instances (%)	Feasible Solutions Gap (%)
J30(50)DR×10	ECI-ICI	RSPH	46	51.3	100	8.7
	—	RSPH	42	34.5	100	9.1
	ECI	RSPH	42	35.3	100	8.6
	ECI	—	42	36.5	100	23.2
	ICI	RSPH	42	39.1	100	7.4
	ECI-ICI	—	40	26.1	100	17.5
	ICI	—	40	31.9	100	15.4
J60(50)DR×10	ECI	RSPH	6	119.6	100	10.4
	ECI-ICI	RSPH	6	164.0	100	10.3
	—	RSPH	6	274.3	100	10.4
	ICI	RSPH	6	310.3	100	10.6
	ECI	—	6	400.6	100	35
	ECI-ICI	—	4	256.0	100	35.5
	ICI	—	4	456.7	100	35.4
J90(50)DR×10	ECI-ICI	RSPH	0	—	100	14.6
	ECI	RSPH	0	—	100	14.6
	—	RSPH	0	—	100	14.7
	ICI	RSPH	0	—	100	14.7
	ECI	—	0	—	100	30.4
	ICI	—	0	—	100	30.4
	ECI-ICI	—	0	—	100	30.4
BLDR×10	ECI	—	56.4	79.1	100	2.2
	ECI-ICI	—	56.4	79.1	100	2.1
	ICI	—	53.8	76.2	100	2.1
	ECI	RSPH	53.8	79.5	100	2.1
	ECI-ICI	RSPH	53.8	79.2	100	2.1
	—	RSPH	48.7	70.6	100	2.1
	ICI	RSPH	48.7	70.4	100	2.2
CNDR×10	ICI	—	1.8	36.7	100	111.2
	ECI	RSPH	1.8	45.4	100	26.3
	ICI	RSPH	1.8	54.2	100	26.3
	ECI-ICI	—	1.8	55.1	100	112.6
	ECI-ICI	RSPH	1.8	75.9	100	26.7
	—	RSPH	1.8	79.1	100	26.2
	ECI	—	1.8	111.8	100	106.8

TABLE 6.7: *The computational efficiency of cutset inequalities and primal heuristics when considering the newly created problem instances with a larger number of resources.*

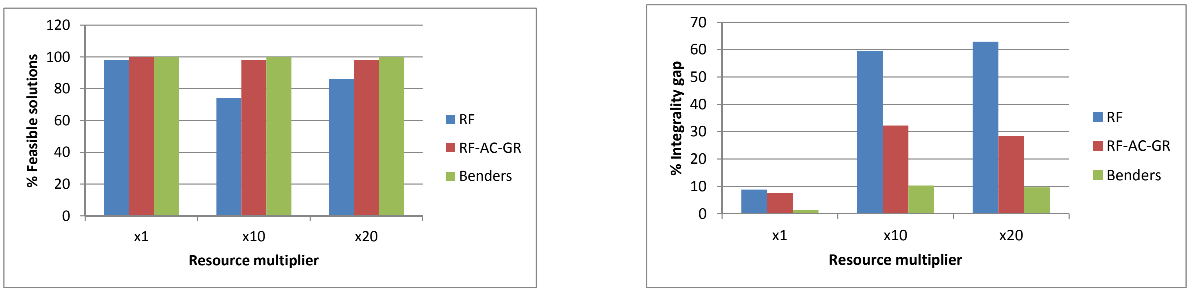
A different picture is portrayed by the results reported in Table 6.5 for the newly created problem instances with a large number of resources. The RFSEP-PDA dominated for the J30(50)DR×10, J60(50)DR×10 and J90(50)DR×10 data sets. For the BLDJ30(50)DR×10 data set, the RFSEP appears to be more efficient and the RFSEP-SLP dominated for the CNDJ30(50)DR×10 data set.

An alarming observation, however, is the large integrality gaps reported for specifically the BLDJ30(50)DR×10 and CNDJ30(50)DR×10 problem instances. For instance, the average integrality gap achieved by means of the RFSEP-AC separation routine for the BLDJ30(50)DR×10 data set is more than 2000%. Very large integrality gaps were also observed for the CNDJ30(50)DR×10 data set. It should be noted, however, that the main objective of generating the results in Tables 6.4 and 6.5 was to evaluate the efficiency of the various separation approaches and to select the most appropriate one to use in further computational tests. For this purpose the RFSEP-PDA was, therefore, selected due to its good performance on specifically the larger and more difficult instances of the J60(50)DR×10 and J90(50)DR×10 data sets.



(a) *Percentage feasible solutions* (b) *Integrality gap*

FIGURE 6.2: *Algorithmic scalability for the J30(50)D data set.*



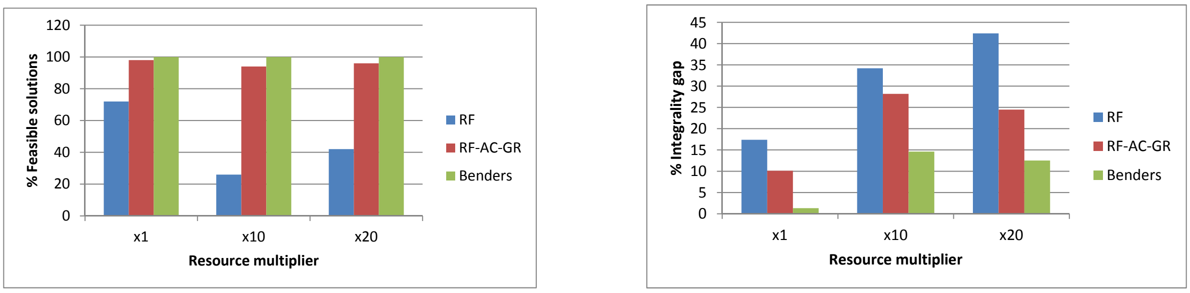
(a) *Percentage feasible solutions* (b) *Integrality gap*

FIGURE 6.3: *Algorithmic scalability for the J60(50)D data set.*

Results on the computational efficiency of the proposed cutset inequalities and primal heuristic are reported in Table 6.6. The performance of the Benders decomposition approach has improved significantly with the inclusion of cutset inequalities and the primal heuristic when compared to Table 6.4. For example, optimal solutions could be computed for 52% of the J30(50)D instances as a consequence of applying the cutset inequalities ECI and ICI, in conjunction with RSPH. Without the cutset inequalities and the primal heuristic, optimal solutions could be computed for only 40% of the instances, according to Table 6.4. The most impressive result, however, is that the corresponding average integrality gap decreased from 193.9% to only 1.7%. Similar results were obtained for the remaining data sets. In most cases, except for the BLD data set, the application of the RSPH appears to improve computing times significantly. There is no clear verdict, however, as to which cutset inequality is dominant.

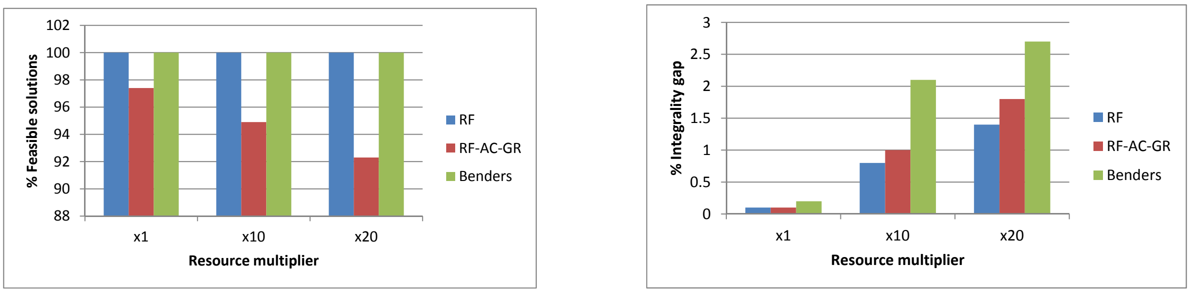
Further results on the proposed cutset inequalities and primal heuristic applied to the newly created data sets which have more resources are reported in Table 6.7. For the larger and more difficult problem instances of the J60(50)DR$\times$10 and J90(50)DR$\times$10 data sets, the combination of the cutset inequalities ECI and ICI appears to be beneficial. The application of the RSPH further improved computing times and it made a significant improvement in reducing the average integrality gap for all but the BLDR$\times$10 problem instances.

The graphs provided in Figures 6.2 to 6.6 serve as a final summary of the computational results obtained for the various algorithmic enhancements presented in this chapter. These graphs show the scalability of the proposed algorithmic enhancements with respect to the various data sets, when the number of resources for each problem instance is increased by a factor of one, ten and twenty, respectively. As an example, to clarify the horizontal axis labels used in the graphs, the results for the "$\times$1" resource multiplier case in Figure 6.2 corresponds to the problem instances of the J30(50)D data set, the results for the "$\times$10" resource multiplier case corresponds to the problem instances of the J30(50)DR$\times$10 data set, and the results for the "$\times$20" resource multiplier case corresponds to the problem instances of the J30(50)DR$\times$20 data set. Figure 6.2(a)



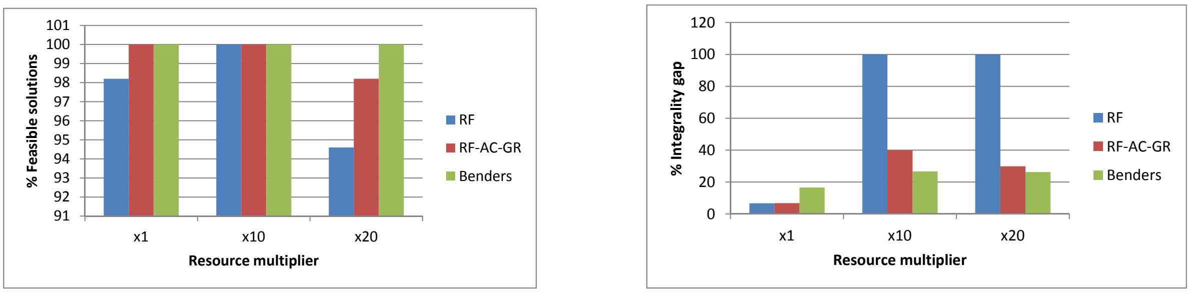
(a) *Percentage feasible solutions* (b) *Integrality gap*

FIGURE 6.4: *Algorithmic scalability for the J90(50)D data set.*



(a) *Percentage feasible solutions* (b) *Integrality gap*

FIGURE 6.5: *Algorithmic scalability for the BLD data set.*



(a) *Percentage feasible solutions* (b) *Integrality gap*

FIGURE 6.6: *Algorithmic scalability for the CND data set.*

shows the percentage of problem instances of the J30(50)D data set for which feasible solutions could be computed by the plain RF problem formulation, the RF-AC-GR (constraint and variable reduction) and Benders decomposition, respectively. The Benders decomposition involves the master problem RFM-GR, the feasibility cut separation problem RFSEP-PDA, the valid inequalities ECI-ICI, and finaly, the primal heuristic RSPH. The average integrality gap obtained for the J30(50)D data set, when considering an increase in the number of resources, is reported by the graph in Figure 6.2(b).

The graphs in Figures 6.2 to 6.6 that report the percentage feasible solutions computed (all the graphs on the left) show that feasible solutions could be computed for all of the problem instances by applying Benders decomposition, even with an increase in the number of resources. By comparing these results to those obtained by the plain RF formulation, the scalability of the Benders approach is clearly demonstrated. This is especially true for the larger and more difficult data sets like the J90(50)D and the CND. Promising results are also reported for the variable and constraint reduction approaches, and very good scalability is maintained for most of the problem instances.

A significant reduction in average integrality gaps were observed for most of the data sets by employing Benders decomposition. The scalability of the Benders decomposition approach, measured in terms of average integrality gap with an increase in the number of resources, is clearly demonstrated by the results for the the larger and more difficult data sets like the J60(50)D, J90(50)D and CND data sets. For these data sets, the average integrality gap obtained through Benders decomposition almost remained constant with an increase in the number of resources. The performance of the plain RF formulation, on the other hand, degraded significantly. Although promising results are reported for the use of the variable and constraint reduction approaches, they are still dominated by the superior performance of Benders decomposition.

6.4 Summary

The primary objective of this chapter was to propose algorithmic approaches for improving computing times when solving the RCSP. Constraint aggregation and graph reduction approaches were presented for reducing the number of variables and constraints in the RF formulation. A branch-and-cut approach was suggested with several sub-problem variations, which are required for separating feasibility cuts, as well as valid inequalities and a heuristic approach for generating primal solutions. The motivation for these algorithmic approaches is based on the observation that typical mine scheduling optimisation problem instances include a large number of resources. Any solution approach employed in a practical mining environment should, therefore, be scalable with an increase in the number of resources.

Computational tests were based on randomly generated data sets with a large number of resources. The results obtained for the various data sets demonstrated the superior scalability of the Benders decomposition approach over adopting the plain RF formulation. The efficiency achieved when solving the RCSP within an exact modelling framework paves the way for incorporating other mine-specific modelling requirements. The next chapter is devoted to modelling extensions and the necessary algorithmic modifications in order to maintain scalability.

CHAPTER 7

Modelling Extensions

Contents

The ability to solve realistic instances of the underground mine scheduling optimisation problem within an exact mathematical programming framework, paves the way for accommodating mining-specific modelling requirements. The aim of this chapter is to present modifications to the RF model of Chapter 4 which address some of these modelling requirements.

Of specific interest is the tracking of crew and equipment movement for the purpose of introducing transfer delays and costs. This is a very relevant topic in current mining conversations due to the increased use of mechanisation [70]. Large machines are used for excavation and these machines have to be scheduled effectively in order to minimise the time it takes to transfer them from one location to the other.

The second requirement to be addressed is the practice of selective mining. Some mining areas may be left behind and discarded permanently due to economic considerations. In order to address this requirement, multi-mode activity scheduling is considered. Each mode of an activity is associated with different costs and resource requirements. Therefore, one of the “modes” may correspond to the decision not to execute a certain mining activity, thus incurring no costs and not consuming or producing any resources.

As a final contribution of this chapter the modelling of uncertainty in resource production and consumption is addressed. The randomness inherent in the resource requirements of mining activities may be attributed, for instance, to the variability in geology or to the inability to excavate underground areas exactly as planned.

The computational results presented in this chapter are only for the objective of maximising NPV. This is motivated by the importance of using NPV as a performance measure in the mining industry for evaluating competing scheduling solutions.

7.1 Crew requirement constraints and transfer delays

The requirement to incorporate the movement of crews and equipment into the RCSP framework is for the purposes of tracking these resources and introducing transfer delays. In the work of Krüger and Scholl [56] and of Quilliot and Toussaint [81], transfer delays are explicitly considered and incorporated into resource flow formulations of the problem. It is shown below that the RF formulation considered in this dissertation for solving the underground mine scheduling problem may easily be extended to facilitate transfer delay constraints on resources.

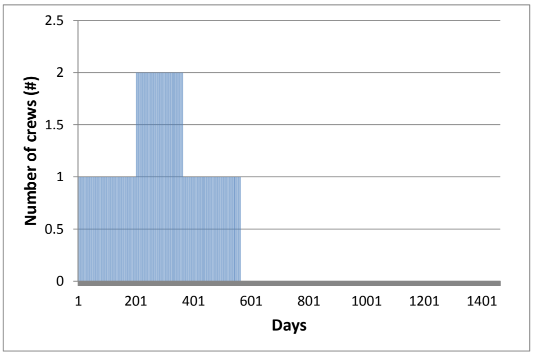
In order to illustrate this more concretely, consider the small mine scheduling example of Section 3.2, for which the mine layout is depicted in Figure 3.5. In order to incorporate crew scheduling, the input data are augmented with crew requirements, as specified in Table 7.1.

				Resource quantities (v_{ir})	
i	Type	d_i	$\mathcal{P}(i)$	Development crew(#)	Stoping crew(#)
1	Haulage	200	-	200	0
2	Cross-cut	64	1	64	0
3	Step-over	10	2	10	0
4	Raise line	90	3	90	0
5	Stope panel	225	4	0	225
6	Stope panel	225	5	0	225
7	Stope panel	225	6	0	225
8	Stope panel	225	7	0	225
9	Haulage	200	1	200	0
10	Cross-cut	64	9	64	0
11	Step-over	10	10	10	0
12	Raise line	90	11	90	0
13	Stope panel	225	12	0	225
14	Stope panel	225	13	0	225
15	Stope panel	225	14	0	225
16	Stope panel	225	15	0	225

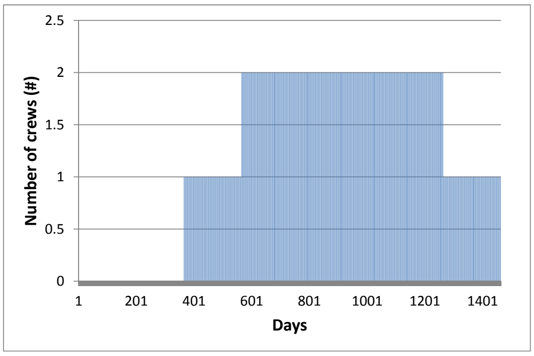
TABLE 7.1: *Activity input values for the small example considered.*

The column labeled "Development crew(#)", specifies the crew requirement for each development activity. Consider the development crew requirement for activity $i = 1$ which is 200. Since the duration of this activity is 200 days, the development requirement for activity i is, therefore, one crew per day. This is exactly in line with the formulation of the resource flow requirement constraints (4.9)–(4.8) in the case of the RF-MIN problem and (4.34)–(4.35) in the case of the RF-MAX problem, since the right-hand side values are $v_{ir}/d_i = 200/200 = 1$. The column labeled "Stoping crew(#)", specifies the crew requirement for each stoping activity and the same logic in terms of the resource requirement constraints is applied for this resource.

For the purpose of illustrating the crew scheduling feature, an unconstrained version of the RF-MAX problem is solved using the data provided in Table 7.1. That is, the original capacity constraint described in Section 3.2 is omitted. The crew requirements for the unconstrained



(a) *Development crew requirements.*



(b) *Stoping crew requirements.*

FIGURE 7.1: *Crew requirements over time for the unconstrained RF-MIN problem.*

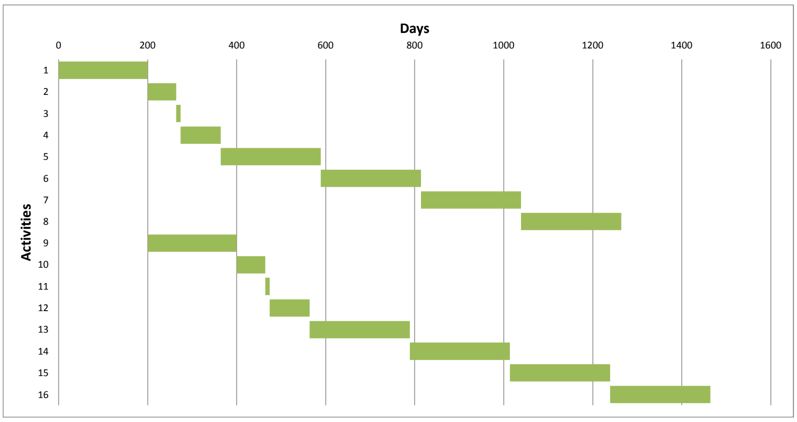


FIGURE 7.2: *Gantt chart representation of a solution to the unconstrained RF-MIN problem.*

version of the RF-MAX problem are provided in Figure 7.1. For both development and stoping, the crew requirements vary mostly between one and two crews per day. These results are confirmed by the Gantt chart in Figure 7.2 showing how activities overlap with the result that more than one crew may be required per day.

For the sake of illustration, assume that there are three crews available per day for this hypothetical mining operation — two crews for development activities and one for stoping activities. In terms of the RF-MAX problem formulation, the upper bound U_r is set equal to two for the development crew resource, and for the stoping crew resources, it is set equal to one. The Gantt chart in Figure 7.3 shows the new schedule obtained by re-solving the RF-MAX problem with the newly added resource constraints. In order to adhere to the constraint of having only one stope crew per day, the first stoping panel from the first raise line, that is activity $i = 5$, is followed by the stoping panels from the second raise line, that is, activities $i = 13$, $i = 14$, $i = 15$ and $i = 16$. Only after completion of the stoping panels in the second raise line are the remaining stoping panels in the first raise line completed, that is, activities $i = 6$, $i = 7$ and $i = 8$. This implies that work on the first raise line is temporarily suspended to start work on the second raise line, which provides access to the stoping panels having a higher mineral content. Only when the stoping panels in the second raise line have been completed is the excavation of the stoping panels in the first raise line resumed. This solution suggests an interrupted work-flow which poses a logistical problem. Every time a mining crew needs to be relocated to a different workplace, equipment has to be moved and possibly be reconfigured.

In order to prevent the undesirable behaviour of temporarily suspending a work place and only

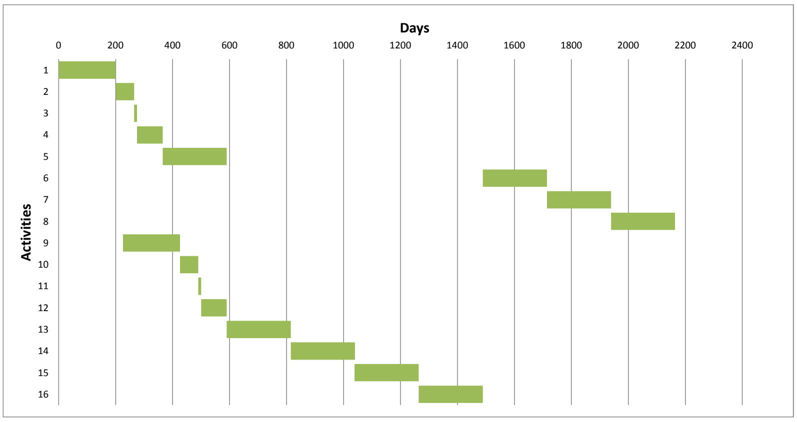


FIGURE 7.3: *Gantt representation of a solution to the crew constrained RF-MAX problem.*

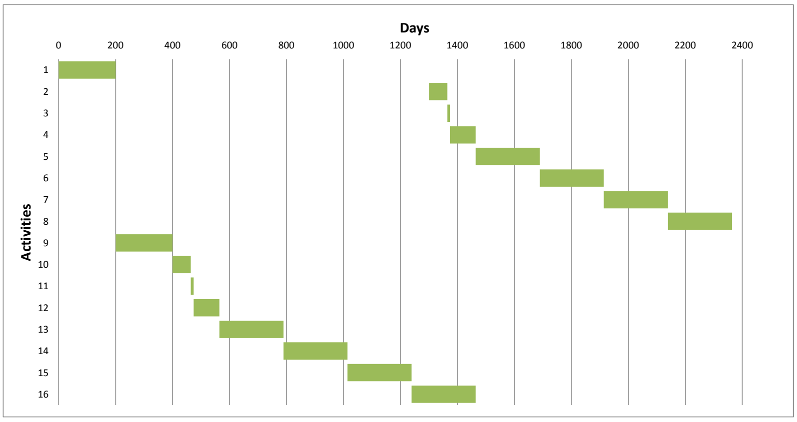


FIGURE 7.4: *Gantt chart representation of a solution to the RF-MAX problem with transfer delay constraints.*

returning to it at a later stage, transfer delays are imposed. By introducing a large transfer delay on the stoping crew when moving from any stoping panel in the first raise line to any stoping panel in the second raise line, the optimisation model is encouraged to provide a solution whereby an entire raise line is completed before commencing work on an adjacent raise line. The Gantt chart in Figure 7.4 shows an optimal solution while taking into account the proposed transfer delays. The required changes to the RF-MAX problem formulation to accommodate transfer delays are provided in the section below. From the Gantt chart it is clear that the optimal solution now suggests a schedule where work on the first raise line is only started once the second raise line has been completed entirely.

7.1.1 Model formulation

The modelling of transfer delay constraints is naturally facilitated by the RF problem formulation. This is true for both the RF-MIN and RF-MAX problem formulations. Consider the transfer delay parameter δ_{ijr}, which specifies the number of days it will take to transfer a resource $r \in \mathcal{R}$ from activity $i \in \mathcal{N}$ to activity $j \in \mathcal{N}$. The formulation of transfer delay constraints involves adapting some of the decision variables and existing constraints in the RF formulations. For instance, an extra index is added to the linear ordering variables and all the constraints that involve these variables are changed accordingly. More specifically, the linear ordering variable is redefined as z_{ijr} and takes on a value of one if resource $r \in \mathcal{R}$ is allowed to be transferred from activity i to j.

The constraints that are affected by this change include the linear ordering constraints (4.7) and (4.33), as well as the constraints (4.10) and (4.36) which permit flow of resources based on the linear ordering variables, in the RF-MIN and RF-MAX problem formulations, respectively. For the sake of completeness the adapted version of the RF-MAX problem formulation is presented below.

The objective of the *resource flow-based RCSP with resource transfer delay constraints, when considering maximisation of NPV* (RF-TD-MAX), is to

$$\text{maximise} \sum_{i \in \mathcal{N}} y_i, \tag{7.1}$$

subject to the constraints

$$\begin{aligned}
s_i - s_j &\leq -d_i, && (i,j) \in \mathcal{Z}, && (7.2)\\
s_j - s_i - (d_i + \delta_{ijr} + M) z_{ijr} &\geq -M, && (i,j) \in \mathcal{A},\ r \in \mathcal{R}, && (7.3)\\
\sum_{(i,j) \in \mathcal{A}(i)} f_{ijr} &= v_{ir}/d_i, && i \in \mathcal{N},\ r \in \mathcal{R}, && (7.4)\\
\sum_{(i,j) \in \mathcal{A}(j)} f_{ijr} &= v_{jr}/d_j, && j \in \mathcal{N},\ r \in \mathcal{R}, && (7.5)\\
f_{ijr} - \min\{v_{ir}/d_i, v_{jr}/d_j\} z_{ijr} &\leq 0, && (i,j) \in \mathcal{A},\ r \in \mathcal{R}, && (7.6)\\
s_i - \sum_{v \in \mathcal{V}} \lambda_{iv} s_{iv} &= 0, && i \in \mathcal{N}, && (7.7)\\
y_i - \sum_{v \in \mathcal{V}} \lambda_{iv} y_{iv} &= 0, && i \in \mathcal{N}, && (7.8)\\
\sum_{v \in \mathcal{V}} \lambda_{iv} &= 1, && i \in \mathcal{N}, && (7.9)\\
\lambda_{i0} - \ell_{i1} &\leq 0, && i \in \mathcal{N}, && (7.10)\\
\lambda_{iv} - \ell_{iv} - \ell_{i(v+1)} &\leq 0, && i \in \mathcal{N},\ v \in \mathcal{V} \setminus \{0, V-1\}, && (7.11)\\
\lambda_{i(V-1)} - \ell_{i(V-1)} &\leq 0, && i \in \mathcal{N}. && (7.12)
\end{aligned}$$

The addition of the explicit precedence constraints (7.2) and the reformulation of constraints (7.3) and (7.6) to accommodate the new linear ordering variables z_{ijr}, are the only changes required in order to obtain the RF-TD-MAX formulation from the RF-MAX formulation. The constraints (7.7)–(7.12) are copies of the constraints (4.37)–(4.42), which are responsible for facilitating the maximisation of NPV.

A direct computational consequence of the above reformulation is that constraint reduction through the use of aggregated constraints, as proposed in Section 6.1, is no longer feasible. The constraints (7.6) cannot be aggregated over the set of resources due to the linear ordering variables that are now also indexed by a resource. It is still possible, however, to make use of graph reduction in order to reduce the number of variables and constraints in the above formulation.

7.1.2 Algorithmic modifications

Although transfer delay constraints are easily incorporated into the RF formulation, several algorithmic modifications are required in order to accommodate the changes within a branch-and-cut framework. The necessary modifications are mainly driven by the fact that the linear ordering variables are now also indexed by a resource identifier.

Generating initial feasible solutions

In order for the initial solutions generated by the TRSH to be feasible, while taking into account transfer delays, a small change to the TRSH algorithm is required. Recall from Section 4.5.1 that as a sub-problem of the TRSH algorithm, the multi-knapsack problem (4.62)–(4.64) is solved iteratively. In order for the TRSH to take transfer delays into account, the precedence constraints (4.63) in the multi-knapsack problem are replaced with the constraints

$$s_j - s_i \geq d_i + \max_{r \in \mathcal{R}}\{\delta_{ijr}\},\ i \in \mathcal{N}^p,\ j \in \mathcal{N}^p. \tag{7.13}$$

The new precedence constraints (7.13) ensure that an activity in the eligibility set $\mathcal{N}^B$ may only start once its predecessors in the processed set $\mathcal{N}^P$ have been completed, while taking transfer delays into account. The rest of the TRSH remains the same and no additional changes are required in Algorithm 4.1.

Deriving Benders feasibility cuts

The changes proposed to the Benders sub-problems, responsible for separating feasibility cuts, are limited to the RFSEP-PDA problem only. This is due to the successes achieved with the RFSEP-PDA problem formulation in the computational tests that were performed on the various problem instances.

Consider the RF-TD-MAX problem formulated according to (7.1)–(7.12). Applying Benders decomposition will result in a master problem that corresponds to the RF-TD-MAX problem formulation, but without the constraints (7.4)–(7.6). These constraints form part of the flow separation problem that is decomposed into independent sub-problems. More, specifically, for each $r \in \mathcal{R}$, the objective of the separation sub-problem is to

$$\text{minimise } \alpha_r, \tag{7.14}$$

subject to the constraints

$$\sum_{(i,j)\in\mathcal{A}(i)} f_{ijr} = v_{ir}/d_i, \qquad i \in \mathcal{N}, \tag{7.15}$$

$$\sum_{(i,j)\in\mathcal{A}(j)} f_{ijr} = v_{jr}/d_j, \qquad j \in \mathcal{N}, \tag{7.16}$$

$$f_{ijr} + \alpha_r \leq \min\{v_{ir}/d_i, v_{jr}/d_j\} z^*_{ijr}, \qquad (i,j) \in \mathcal{A}. \tag{7.17}$$

If $\alpha_r > 0$ for any $r \in \mathcal{R}$, then the current vector $\boldsymbol{z}^*$ is infeasible. The separation problem (7.14)–(7.17) is exactly the same as the RFSEP-DA separation problem (6.32)–(6.35), except that an additional index is present for the linear ordering variables $\boldsymbol{z}^*$.

Theorem 7.1. *For a given $r \in \mathcal{R}$ and a vector $\boldsymbol{z}^* \in \{0,1\}^{|\mathcal{A}|\times|\mathcal{R}|}$, the cut*

$$\sum_{(i,j)\in\mathcal{A}} \min\{v_{ir}/d_i, v_{jr}/d_j\} z_{ijr}\mu_{ij} \leq -\sum_{i\in\mathcal{N}} v_{ir}/d_i \left(\pi_i^1 + \pi_i^2\right), \tag{7.18}$$

where $\boldsymbol{\pi}^1 \in \mathbb{R}^{|\mathcal{N}|}$, $\boldsymbol{\pi}^2 \in \mathbb{R}^{|\mathcal{N}|}$ and $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{A}|}_{\leq 0}$ are the dual vectors associated with (7.15), (7.16) *and* (7.17)*, respectively, separates the infeasible point $\boldsymbol{z}^*$ if $\alpha_r > 0$.*

Proof. The proof is analogous to the proof of Theorem 6.3 with the exception that the linear ordering variables have an additional index for the resource identifier. □

A parallel implementation of the feasibility cuts (7.56) is obtained by allowing the separation problem (7.14)–(7.17) to be solved in parallel for each resource $r \in \mathcal{R}$.

Valid inequalities

The explicit and implicit cutset inequalities proposed in Section 6.2.10, are modified to remain valid when transfer delays are considered.

Proposition 7.1. *For a specific resource $r \in \mathcal{R}$ and a subset $\mathcal{V}$, the valid inequality*

$$\sum_{(i,j)\in\mathcal{A}(\mathcal{V},\bar{\mathcal{V}})} \min\{v_{ir}/d_i, v_{jr}/d_j\} z_{ijr} \;\geq\; \max_{i\in\mathcal{V}}\{v_{ir}/d_i\} \tag{7.19}$$

must hold in order for any given solution $\boldsymbol{z}^*$ to be feasible.

Proof. The proof is analogous to the proof of Theorem 6.1. □

Proposition 7.2. *For any subset $\mathcal{V}$, the valid inequality*

$$\sum_{(i,j)\in\mathcal{A}(\mathcal{V},\bar{\mathcal{V}})} \sum_{r\in\mathcal{R}} z_{ijr} \geq |\mathcal{V}| \tag{7.20}$$

must hold in order for any given solution $\boldsymbol{z}^*$ to be feasible.

Proof. The proof is analogous to the proof of Theorem 6.1. □

As part of the separation routine responsible for generating the implicit cutset inequalities (7.20), a maximum flow problem is solved for each resource $r \in \mathcal{R}$ in order to identify a minimum cut. Let z^*_{ijr} be the LP relaxation solution for the RFM at a branch-and-bound node, for all $(i,j) \in \mathcal{A}$ and all $r \in \mathcal{R}$. By employing the preflow–push algorithm of Goldberg and Tarjan [40] for each $r \in \mathcal{R}$ with z^*_{ijr} as the arc capacities, a minimum cut $\mathcal{A}(\mathcal{V}, \bar{\mathcal{V}})$ is obtained and used to derive the valid inequality (7.20).

Generating primal feasible solutions

A small modification is required to the RSPH, introduced in Section 6.2.11, in order to generate primal feasible solutions that satisfy transfer delay constraints. Recall from the algorithm outline of the RSPH, given in Algorithm 6.1, that the set $\mathcal{N}^B$ comprises activities that are eligible to start during the current time period $t \in \mathcal{T}$. The set $\mathcal{N}^P$ is introduced for activities already processed and for which start time values s^*_i have been determined. Once an activity i is selected to start during a time period t, its start time is set to $s^*_i = S^{\mathcal{T}}_t$, it is added to $\mathcal{N}^P$ and it is removed from $\mathcal{N}^B$. An activity j may only be considered eligible to start during a time period t if all of its predecessors i have completed prior to period t, while taking transfer delays into account. More specifically, $\mathcal{N}^B = \mathcal{N}^B \cup \{j\}$ only if $i \in \mathcal{N}^P$ for all $(i,j) \in \mathcal{Z}$, and only if $s_j \geq s_i + d_i + \max_{r\in\mathcal{R}}\{\delta_{ijr}\}$.

		Solved to optimality		Feasible Solutions	
Data set	Model	Instances (%)	Time (s)	Instances (%)	Gap (%)
J30(50)DR×10_TD	RF-GR	32	5.1	100	43.3
	RF	28	5.4	92	87.4
	Benders	8	78.9	100	3.7
J60(50)DR×10_TD	RF	4	499.7	24	7.6
	Benders	0	—	100	3.9
	RF-GR	0	—	100	36.6
J90(50)DR×10_TD	Benders	0	—	100	5.2
	RF-GR	0	—	96	35.5
	RF	0	—	4	24
BLDR×10_TD	RF	12.8	19.5	97.4	10.8
	RF-GR	12.8	25.1	92.3	10.5
	Benders	0	—	100	7.2
CNDR×10_TD	Benders	0	—	100	12.4
	RF-GR	0	—	94.6	241.4
	RF	0	—	73.2	781.3

TABLE 7.2: *The effect of applying resource transfer delays on computing times.*

7.1.3 Computational results

For the purpose of testing the effect that transfer delay constraints have on computing times, the newly created data sets introduced earlier are augmented with information on randomly generated transfer delays. More specifically, for each resource $r \in \mathcal{R}$, the delay in transferring the resource from activity $i \in \mathcal{N}$ to $j \in \mathcal{N}$, is given by $\delta_{ijr} \sim U(0, 10)$. The name of each data set augmented with transfer delays is given a post-fix "TD."

The results obtained when solving the RCSP with transfer delay constraints are listed in Table 7.2. Optimal solutions were computed for the J30(50)DR×10_TD and the BLDR×10_TD data sets. Although Benders decomposition does not appear to perform well in terms of optimal solutions computed, feasible solutions were computed for all of the problem instances in all of the data sets. Furthermore, the average integrality gap obtained by applying Benders decomposition is significantly lower than in solutions to the plain RF formulation and the RF formulation with graph reduction. For instance, the integrality gap for the BLDR×10_TD data set when applying Benders decomposition is on average 12.4%, compared to 781.3% and 241.4% when applying the plain RF formulation and the RF formulation with graph reduction, respectively.

7.2 Selective scheduling

Selective scheduling is only applicable to the case where maximisation of NPV is considered. This is especially true within an underground mining environment where high variability in the mineral deposits may prompt selective excavation of parts of the mine in order to promote profitability. Selective scheduling should still adhere to precedence constraints and, therefore, the most benefit is derived within a mining context when low-grade mining areas do not precede high-grade mining areas.

A very important application of selective mining is the evaluation of future capital projects. In an underground mine, this may involve the sinking of an additional shaft that allows access to potential profitable mining areas. The financial trade-off between the capital costs required for sinking the shaft and the potential revenue that can be generated, may now be incorporated directly into the RCSP model.

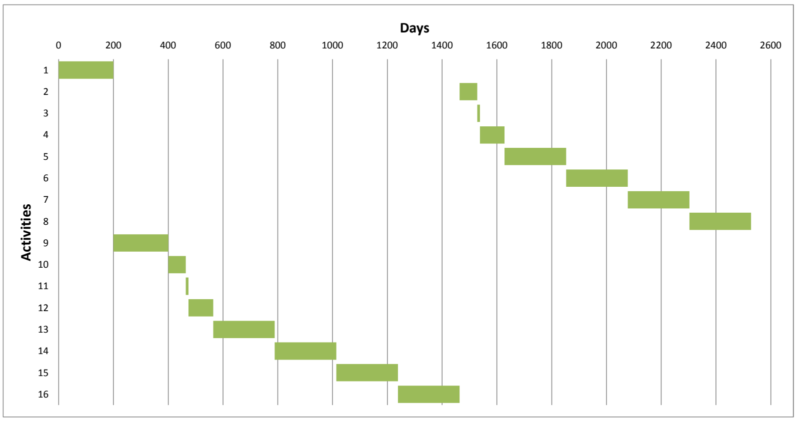


FIGURE 7.5: *Gantt chart representation of a solution to the RF-MAX problem when stoping activities $i = 5$, $i = 6$, $i = 7$ and $i = 8$ are assumed to have no mineral content.*

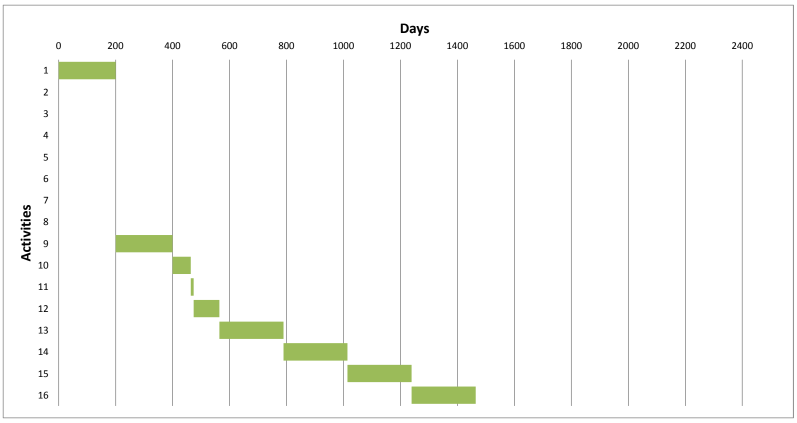


FIGURE 7.6: *Gantt chart representation of a solution to the RF-MAX problem when considering selective scheduling. The stoping activities $i = 5$, $i = 6$, $i = 7$ and $i = 8$ are assumed to have no mineral content.*

For illustrative purposes consider the small mine example from Section 3.2 and the results obtained for the unconstrained RF-MAX provided in Figure 7.2. The solution for the unconstrained RF-MAX suggests the excavation of all the stoping panels as soon as possible. By doing this a maximum NPV is obtained. Furthermore, even though the development activities do not generate any revenue, they are still required to be scheduled in order to give access to the revenue generating stoping activities.

Consider the hypothetical case where the stoping panels in the first raise line, *i.e.* $i = 5$, $i = 6$, $i = 7$ and $i = 8$, do not contain any mineral value. By resolving the RF-MAX problem, a solution is obtained for which all the activities associated with the first raise line are scheduled as late as possible, see Figure 7.5. This is expected since it is unprofitable to excavate these "empty" stoping panels and in order to maximise NPV, all these costly activities are scheduled to start as late as possible.

In order to prevent that unprofitable activities are scheduled as part of an optimal solution, selective scheduling is incorporated into the RF-MAX formulation. The result for the above example is that a solution is provided which does not schedule the "empty" stoping activities or any of their predecessors. The only activities that are scheduled correspond to those activities that are associated with the second raise line and for which the stoping panels have mineral content. The corresponding Gantt chart for this solution is depicted in Figure 7.6.

7.2.1 Model formulation

The RF formulation requires additional variables and constraints in order to accommodate selective scheduling. Denote by $x_i \in \{0,1\}$ the option to schedule an activity $i \in \mathcal{N}$. That is, if $x_i = 1$ in a feasible solution, then a starting solution s_i is calculated which is feasible in terms of all the constraints in the RCSP formulation. Furthermore, if $x_i = 1$, the revenue or cost associated with activity i is taken into account in the calculation of the objective function.

The introduction of the selective scheduling variable x_i affects several constraints and the entire model is presented below for the sake of completeness. The objective of the *resource flow-based RCSP with selective scheduling, when considering maximisation of NPV* (RF-SEL-MAX), is to

$$\text{maximise} \sum_{i \in \mathcal{N}} y_i, \tag{7.21}$$

subject to the constraints

$$\begin{aligned}
x_j - x_i &\leq 0, && (i,j) \in \mathcal{Z},\ j \neq \{N\}, && (7.22)\\
s_j - s_i - (d_i + M)z_{ij} &\geq -M, && (i,j) \in \mathcal{A}, && (7.23)\\
\sum_{(i,j)\in\mathcal{A}(i)} f_{ijr} - v_{ir}/d_i x_i &= 0, && i \in \mathcal{N},\ r \in \mathcal{R}, && (7.24)\\
\sum_{(i,j)\in\mathcal{A}(j)} f_{ijr} - v_{jr}/d_j x_j &= 0, && j \in \mathcal{N},\ r \in \mathcal{R}, && (7.25)\\
f_{ijr} - \min\{v_{ir}/d_i, v_{jr}/d_j\}z_{ij} &\leq 0, && (i,j) \in \mathcal{A},\ r \in \mathcal{R}, && (7.26)\\
s_i - \sum_{v\in\mathcal{V}} \lambda_{iv}s_{iv} &= 0, && i \in \mathcal{N}, && (7.27)\\
y_i - \sum_{v\in\mathcal{V}} \lambda_{iv}y_{iv} + Mx_i &\leq M, && i \in \mathcal{N}, && (7.28)\\
-y_i + \sum_{v\in\mathcal{V}} \lambda_{iv}y_{iv} + Mx_i &\leq M, && i \in \mathcal{N}, && (7.29)\\
\sum_{v\in\mathcal{V}} \lambda_{iv} &= 1, && i \in \mathcal{N}, && (7.30)\\
\lambda_{i0} - \ell_{i1} &\leq 0, && i \in \mathcal{N}, && (7.31)\\
\lambda_{iv} - \ell_{iv} - \ell_{i(v+1)} &\leq 0, && i \in \mathcal{N},\ v \in \mathcal{V}\setminus\{0, V-1\}, && (7.32)\\
\lambda_{i(V-1)} - \ell_{i(V-1)} &\leq 0, && i \in \mathcal{N}. && (7.33)
\end{aligned}$$

The newly introduced constraint set (7.22) is required in order to ensure that an activity $j \in \mathcal{N}$ may only be selected for scheduling if its predecessor $i \in \mathcal{N}$ has been selected for scheduling as well. The only exception is that the sink node N is allowed to be scheduled even if its predecessors are not scheduled, since the sink node does not incur any cost or revenue.

Constraint sets (7.24) and (7.25) are the modified resource flow balancing constraints. These constraints dictate that resource requirements are only enforced for an activity i if the activity is selected for scheduling through the selective scheduling variable x_i.

The constraint sets (7.28) and (7.29) are necessary to calculate the objective function contribution by each of the activities, based on the selective scheduling variable x_i. That is, if $x_i = 1$, the revenue or cost associated with activity i is assigned to the variable y_i. If, on the other hand, $x_i = 0$, then $y_i = 0$.

7.2.2 Algorithmic modifications

The proposed changes to the RF formulation for the purpose of accommodating selective scheduling do not require any modifications to the TRSH. A feasible solution generated by the TRSH in its current form implies that all activities are selected for scheduling. The computational results that follow demonstrate that improved computing times are still observed when applying the TRSH in conjunction with the RF-SEL-MAX problem formulation. Furthermore, constraint and variable reduction achieved through constraint aggregation and graph reduction (as introduced in Section 6.1), are not affected by the newly introduced formulation and may still contribute to improving computing times.

The introduction of the constraints (7.22), (7.28) and (7.29) only affects the master problem formulation of the Benders decomposition. The functioning of the primal heuristic RSPH (introduced in Section 6.2.11) and the application of the cutset inequalities (introduced in Section 6.2.10) are, therefore, unaffected. Changes are, however, required to the Benders feasibility sub-problem due to the introduction of the selective scheduling variables x_i into constraints (7.24) and (7.25).

Deriving Benders feasibility cuts

The proposed changes to the Benders sub-problem for the purpose of incorporating selective scheduling are restricted to the RFSEP-PDA due to its computational efficiency.

Consider the RF-SEL-MAX problem formulation (7.21)–(7.33). Applying Benders decomposition will result in a master problem that corresponds to the RF-SEL-MAX problem formulation, but without the constraints (7.24)–(7.26). These constraints form part of the flow separation problem that is decomposed into independent sub-problems. Let $\boldsymbol{x}^*$ be the current solution in the Benders master problem corresponding to the selective scheduling variables and let $\boldsymbol{z}^*$ be the current solution of the linear ordering variables. Then, for each $r \in \mathcal{R}$, the objective of the separation sub-problem is to

$$\text{minimise } \alpha_r, \tag{7.34}$$

subject to the constraints

$$\begin{aligned}
\sum_{(i,j)\in\mathcal{A}(i)} f_{ijr} &= v_{ir}/d_i x_i^*, && i \in \mathcal{N}, && (7.35)\\
\sum_{(i,j)\in\mathcal{A}(j)} f_{ijr} &= v_{jr}/d_j x_j^*, && j \in \mathcal{N}, && (7.36)\\
f_{ijr} + \alpha_r &\leq \min\{v_{ir}/d_i, v_{jr}/d_j\} z_{ij}^*, && (i,j) \in \mathcal{A}. && (7.37)
\end{aligned}$$

If $\alpha_r > 0$ for any $r \in \mathcal{R}$, then the current vectors $\boldsymbol{x}^*$ and $\boldsymbol{z}^*$ are infeasible. The separation problem (7.34)–(7.37) is exactly the same as the RFSEP-DA separation problem (6.32)–(6.35), except for the right-hand-side of constraints (7.35) and (7.36), which are multiplied by the solution values x_i^*.

Theorem 7.2. *For a given $r \in \mathcal{R}$, $\boldsymbol{x}^* \in \{0,1\}^{|\mathcal{N}|}$ and $\boldsymbol{z}^* \in \{0,1\}^{|\mathcal{A}|}$, the cut*

$$\sum_{(i,j)\in\mathcal{A}} \min\{v_{ir}/d_i, v_{jr}/d_j\} z_{ij}\mu_{ij} + \sum_{i\in\mathcal{N}} v_{ir}/d_i \left(\pi_i^1 + \pi_i^2\right) x_i \leq 0, \tag{7.38}$$

where $\boldsymbol{\pi}^1 \in \mathbb{R}^{|\mathcal{N}|}$, $\boldsymbol{\pi}^2 \in \mathbb{R}^{|\mathcal{N}|}$ and $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{A}|}_{\leq 0}$ are the dual vectors associated with (7.35), (7.36) *and* (7.37)*, respectively, separates the infeasible point defined by $\boldsymbol{x}^*$ and $\boldsymbol{z}^*$, if $\alpha_r > 0$.*

Proof. The proof is analogous to the proof of Theorem 6.3. □

		Solved to optimality		Feasible Solutions	
Data set	Model	Instances (%)	Time (s)	Instances (%)	Gap (%)
J30(50)DR×10	Benders	74	29.5	100	1.5
	RF	56	28.7	100	5.3
	RF-AC-GR	54	22.1	100	4.3
J60(50)DR×10	Benders	26	137.0	100	9
	RF-AC-GR	12	239.7	100	25.9
	RF	10	177.4	64	450.4
J90(50)DR×10	Benders	12	471.1	100	13.7
	RF-AC-GR	0	—	98	29
	RF	0	—	46	5276.7
BLDR×10	Benders	97.4	27.8	100	0.2
	RF-AC-GR	94.9	16.2	100	0.3
	RF	92.3	11.2	100	0.3
CNDR×10	Benders	48.2	25.8	100	3.8
	RF	32.1	62.9	100	6.6
	RF-AC-GR	32.1	71.3	100	5.4

TABLE 7.3: *The effect of applying selective scheduling on computing times.*

7.2.3 Computational results

In order to determine what the effect is on computing times when considering selective scheduling, no additional data are required. The data sets introduced in earlier chapters suffice for this purpose.

The results obtained when solving the RCSP with selective scheduling considerations are listed in Table 7.3. The computational efficiency achieved with Benders decomposition is evident for all of the data sets considered. Apart from the largest percentage of instances solved to optimality, the Benders decomposition approach was also able to yield feasible solutions for all of the problem instances. Significant improvements in the average integrality gap is, furthermore, evident when applying Benders decomposition. For instance, the average integrality gap observed for the J60(50)DR×10 data set when applying Benders decomposition is 9%, compared to the 25.9% and 450% when solving the RCSP with RF-AC-GR and the plain RF formulation, respectively. For the J90(50)DR×10 data set, an average integrality gap of 13.7% is obtained by applying Benders decomposition, compared to 29% and 5 276% when applying RF-AC-GR and the plain RF formulation, respectively.

7.3 Uncertainty in resource requirements

Resource constrained scheduling in the face of uncertainty is mainly achieved according to one of the following approaches: The *reactive scheduling* methodology involves a full or partial re-scheduling in the event of unexpected circumstances [1, 5, 39]. The focus is therefore on "repairing" the baseline schedule rather than constructing a schedule that may be robust enough to accommodate unforeseen circumstances. *Proactive scheduling*, on the other hand, involves the attempted generation of schedules that are feasible for a predefined range of stochastic events which may influence, for instance, the activity durations. In the literature, these types of problem formulations are also referred to as *robust* or *stochastic* scheduling problems. There is, however, a further distinction to be made between two types of proactive scheduling approaches. The first may be classified as a *single-stage* robust or stochastic programming problem and it involves the computation of a single schedule for which different resource allocations apply, depending on the specific realisation of a stochastic event. The papers [28, 29, 44] present various single-stage

		Mineral content (kg)		
i	Type	Low	High	Average
1	Haulage	0	0	0
2	Cross-cut	0	0	0
3	Step-over	0	0	0
4	Raise line	0	0	0
5	Stope panel	30	45	37.5
6	Stope panel	30	45	37.5
7	Stope panel	30	45	37.5
8	Stope panel	30	45	37.5
9	Haulage	0	0	0
10	Cross-cut	0	0	0
11	Step-over	0	0	0
12	Raise line	0	0	0
13	Stope panel	45	45	45
14	Stope panel	45	45	45
15	Stope panel	45	45	45
16	Stope panel	45	45	45

TABLE 7.4: *A scenario representation of uncertainty in mineral content.*

approaches to deal with the RCSP in the face of uncertainty with respect to activity durations. The model proposed by Lambrechts *et al.* [58] is a single-stage problem for computing schedules that are protected against uncertainty in resource availability.

The second type of proactive scheduling involves a *dynamic decision process*. Instead of having an explicit problem formulation, a feasible schedule is created by stepping through time and at each decision point, which typically coincides with completion times of activities, a *policy* is applied to decide on which activities are released for execution. The papers [3, 6, 62] contain various approaches towards solving the RCSP with uncertainty in respect of activity durations.

The approach followed in this thesis towards modelling uncertainty in the resource requirements of the RCSP, is a combination of the two types of proactive scheduling described above. More specifically, a two-stage stochastic programming formulation is adopted (see Higle [45] for a good introduction to this type of mathematical model) whereby the set of activities are partitioned into first-stage and second-stage activities. The first-stage activities are those activities that are typically associated with initial capital layout. For instance, off-reef development activities are only cost generating activities, but they are required in order to provide access to the reef. The property of a first-stage activity is that, for a given solution of an RCSP instance with uncertainty in the resource requirements, it should have the same start time over all of the stochastic scenarios considered. The argument is that a first-stage activity is not directly influenced by uncertainty and its start time should, therefore, be determined proactively with respect to the different stochastic scenarios. The stoping activities, on the other hand, are chosen to be the second-stage activities since they are exposed to uncertainty in respect of how much mineral content will be recovered when excavation takes place. The stoping activities are, furthermore, the profit generating activities in terms of the NPV objective function. The start time of the second-stage activities should be scenario-dependent to allow for robustness in the schedule whenever future stochastic scenarios realises.

For illustration, consider again the small mine example of Section 3.2. Table 7.4 provides a scenario representation of uncertainty in the mineral content found in the stoping panels. Although a “Low” and a “High” scenario are provided, only variability in the mineral content

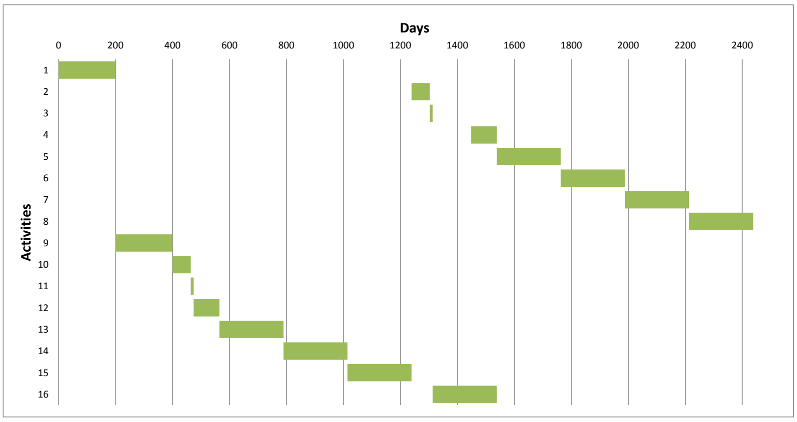


FIGURE 7.7: *Gantt chart representation of a solution to the RF-MAX problem when solving the "Low" scenario.*

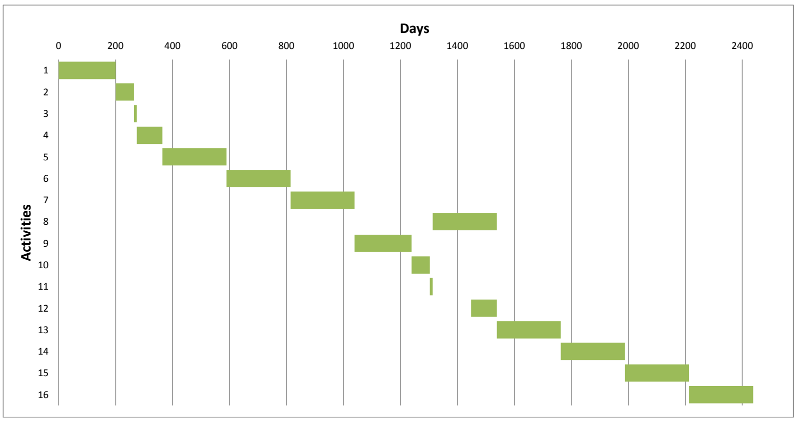


FIGURE 7.8: *Gantt chart representation of a solution to the RF-MAX problem when solving the "High" scenario.*

of the stoping panels from the first raise line is considered. The "Average" column contains the average mineral content over the two scenarios.

A deterministic approach towards utilising the data in Table 7.4 would require solving the RF-MAX problem for the two scenarios separately. To this end, the results obtained by solving the RF-MAX problem for the "Low" scenario and for the "High" are depicted in Figures 7.7 and 7.8, respectively. An important observation is that the results obtained by solving the RF-MAX problem for the "Low" and the "High" scenarios are completely different. The fact that the results for the two scenarios cannot be reconciled to form a single plan is problematic from a management perspective. Although the benefit of having a robust proactive plan is achieved, that is, having a suggested course of action depending on which scenario realises, some certainty is required with respect to the initial capital layout. More specifically, the second haulage of the small mine example, activity $i = 9$, is scheduled to start immediately after the first section of the haulage, activity $i = 1$, when considering the "Low" scenario. But for the "High" scenario, activity $i = 9$ is scheduled to start much later. In practice, the haulage activities are considered to be part of the initial capital layout and within a two-stage stochastic framework these activities are part of the set of first-stage activities. The idea of having a first-stage activity, as mentioned before, is that it should be scheduled to start at the same time for all of the scenarios considered.

The Gantt charts in Figures 7.9 and 7.10, represent the results obtained when solving the RCSP as a two-stage stochastic programming problem. The results in Figure 7.9 show the start times

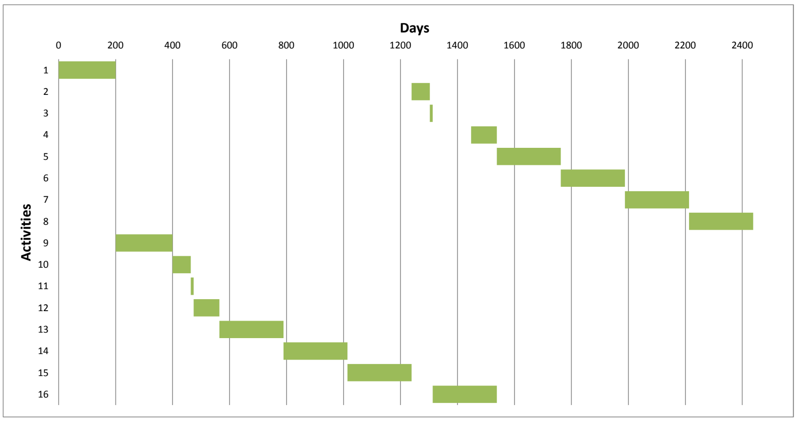


FIGURE 7.9: *Gantt chart representation of a solution to the "Low" scenario when solving a two-stage RCSP with activities $i = 1$ and $i = 9$ as the first-stage activities and all other activities as the second-stage activities.*

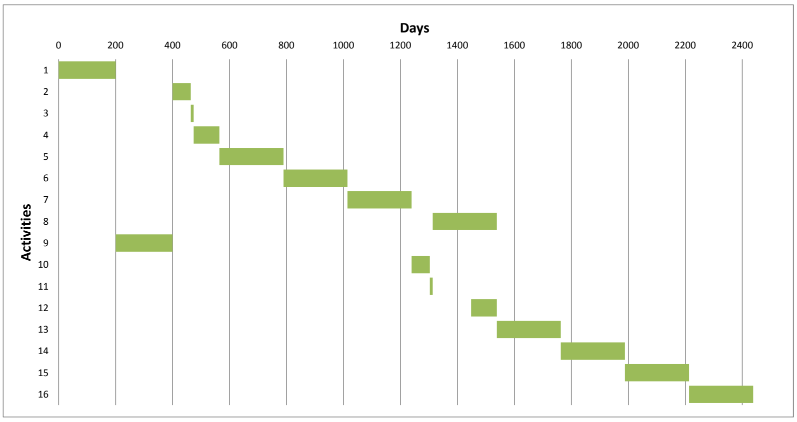


FIGURE 7.10: *Gantt chart representation of a solution to the "High" scenario when solving a two-stage RCSP with activities $i = 1$ and $i = 9$ as the first stage activities and all other activities as the second-stage activities.*

of the activities if the "Low" scenario should realise in future, while the results in Figure 7.10 show the start times of the activities if the "High" scenario should realise in future. Since the haulage activities $i = 1$ and $i = 9$ are considered to be first-stage activities, their start times remain the same irrespective of which scenario will realise in future.

It is possible to generate an entire proactive scheduling solution where all activities only have a single start time solution. This can simply be achieved by solving the RF-MAX problem and making use of the average mineral content over the two scenarios, given by the last column of Table 7.4. The solution to this problem instance corresponds to the Gantt chart for the "Low" scenario depicted in Figure 7.7. Since only one set of starting times for the activities is obtained as a solution, robustness is compromised with the result that the NPV is also lower than the expected NPV achieved by the two-stage stochastic solution.

7.3.1 Model formulation

The model proposed below for solving the RCSP with uncertainty in respect of the resource requirements, is a two-stage stochastic programming formulation based on the resource flow formulation. A scenario-based approach is followed for representing uncertainty and the elements of the set $\mathcal{K} = \{1, 2, \ldots, |\mathcal{K}|\}$ are used to denote the scenario indices. Let d_{ik} be the duration of an activity $i \in \mathcal{N}$ for scenario $k \in \mathcal{K}$ and let v_{irk} be the quantity of resource $r \in \mathcal{R}$ required by activity $i \in \mathcal{N}$, for scenario $k \in \mathcal{K}$.

The set of activities $\mathcal{N}$ is partitioned into first-stage activities $\mathcal{N}^F$ and second-stage activities $\mathcal{N} \setminus \mathcal{N}^F$. Although only the second-stage activities are subjected to uncertainty, a start time variable s_{ik} is introduced for each activity $i \in \mathcal{N}$ and for each scenario $k \in \mathcal{K}$. In order to force the first-stage activities to have a single start time solution which is independent of any scenario, the variable s_i is introduced for each first-stage activity $i \in \mathcal{N}^F$ and the constraint $s_i = s_{ik}$ is imposed for each first-stage activity $i \in \mathcal{N}^F$ and scenario $k \in \mathcal{K}$. The motivation for incorporating the redundant start time variables s_{ik}, which are associated with the first-stage activities $\mathcal{N}^F$, is to simplify the problem formulation. It is assumed that the efficient preprocessing capabilities of most commercial solvers will remove any redundant variables and constraints.

All of the remaining variables in the original RF-MAX formulation are also extended in order to accommodate a scenario representation. The resource flow variables $f_{ijrk} \geq 0$ and the linear ordering variables $z_{ijk} \in \{0, 1\}$ are introduced for each scenario $k \in \mathcal{K}$. The decision variable $y_{ik} \in \mathbb{R}$ denotes the approximate value of the NPV function $f_i(s_{ik})$ for scenario $k \in \mathcal{K}$. The auxiliary variables $\lambda_{ivk} \geq 0$, with $v \in \mathcal{V}$ and $\ell_{ivk} \in \{0, 1\}$ with $v \in \mathcal{V} \setminus \{0\}$ are defined for each $k \in \mathcal{K}$ for the purpose of facilitating the approximation of the objective function. The NPV graph for an activity $i \in \mathcal{N}$ and a scenario $k \in \mathcal{K}$ contains the knots (s_{ivk}, y_{ivk}), for all $v \in \mathcal{V}$.

The objective of the *resource flow-based RCSP with uncertain resource requirements, when considering the maximisation of NPV* (RF-STOCH-MAX) is to

$$\text{maximise} \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} y_{ik}, \tag{7.39}$$

subject to the constraints

$$\begin{aligned}
s_i &= s_{ik}, && i \in \mathcal{N}^F,\ k \in \mathcal{K}, && (7.40)\\
z_{ijk} &= 1, && (i,j) \in \mathcal{Z},\ k \in \mathcal{K}, && (7.41)\\
s_{jk} - s_{ik} - (d_{ik} + M) z_{ijk} &\geq -M, && (i,j) \in \mathcal{A},\ k \in \mathcal{K}, && (7.42)\\
\sum_{(i,j) \in \mathcal{A}(i)} f_{ijrk} &= v_{irk}/d_{ik}, && i \in \mathcal{N},\ r \in \mathcal{R},\ k \in \mathcal{K}, && (7.43)\\
\sum_{(i,j) \in \mathcal{A}(j)} f_{ijrk} &= v_{jrk}/d_{jk}, && j \in \mathcal{N},\ r \in \mathcal{R},\ k \in \mathcal{K}, && (7.44)\\
f_{ijrk} - \min\{v_{irk}/d_{ik}, v_{jrk}/d_{jk}\} z_{ijk} &\leq 0, && (i,j) \in \mathcal{A},\ r \in \mathcal{R},\ k \in \mathcal{K}, && (7.45)\\
s_{ik} - \sum_{v \in \mathcal{V}} \lambda_{ivk} s_{ivk} &= 0, && i \in \mathcal{N},\ k \in \mathcal{K}, && (7.46)\\
y_{ik} - \sum_{v \in \mathcal{V}} \lambda_{ivk} y_{ivk} &= 0, && i \in \mathcal{N},\ k \in \mathcal{K}, && (7.47)
\end{aligned}$$

$$\begin{aligned}
\sum_{v \in \mathcal{V}} \lambda_{ivk} &= 1, && i \in \mathcal{N},\ k \in \mathcal{K}, && (7.48)\\
\lambda_{i0k} - \ell_{i1k} &\leq 0, && i \in \mathcal{N},\ k \in \mathcal{K}, && (7.49)\\
\lambda_{ivk} - \ell_{ivk} - \ell_{i(v+1)k} &\leq 0, && i \in \mathcal{N},\ v \in \mathcal{V} \setminus \{0, V-1\},\ k \in \mathcal{K}, && (7.50)\\
\lambda_{i(V-1)k} - \ell_{i(V-1)k} &\leq 0, && i \in \mathcal{N},\ k \in \mathcal{K}. && (7.51)
\end{aligned}$$

The objective function (7.39) and all of the constraints (7.41)–(7.51) correspond to the objective function and constraints of the RF-MAX problem formulation, except that they are extended to incorporate a scenario representation. The only new constraints that were added are the constraints (7.40), which force every first-stage activity to have the same start time solution across all scenarios.

7.3.2 Algorithmic modifications

Several algorithmic modifications are required to accommodate the two-stage stochastic formulation changes within a branch-and-cut framework. The required changes are simplified since the scenarios in the set $\mathcal{K}$ are assumed to be independent of each other. The only complicating factor is the requirement that the start time solution of a first-stage activity should be the same across all scenarios.

Generating initial feasible solutions

The generation of initial feasible solutions is simplified by employing the TRSH in two stages. During the first stage, the TRSH is applied only to the first-stage activities in order to compute start time solutions s_i for each $i \in \mathcal{N}^F$. During the second stage, the first-stage activities are assigned to the set of processed activities, that is, $\mathcal{N}^P = \mathcal{N}^F$. Start times s_{ik} are computed for each second-stage activity $i \in \mathcal{N} \setminus \mathcal{N}^F$ and for each scenario $k \in \mathcal{K}$, using the TRSH. This is easily accomplished since the scenarios in the set $\mathcal{K}$ are independent of each other.

Deriving Benders feasibility cuts

The proposed changes to the Benders sub-problem for the purpose of incorporating uncertainty in the resource requirements are restricted to the RFSEP-PDA formulation due to its computational efficiency.

Consider the RF-STOCH-MAX formulation (7.39)–(7.51). Applying Benders decomposition will result in a master problem that corresponds to the RF-STOCH-MAX problem formulation, but without the constraints (7.43)–(7.45). These constraints form part of the flow separation problem that is decomposed into independent sub-problems. Let $\boldsymbol{z}^*$ be the current solution in the Benders master problem corresponding to the linear ordering variables. Then, for each $k \in \mathcal{K}$ and each $r \in \mathcal{R}$, the objective of the separation sub-problem is to

$$\text{minimise } \alpha_{rk}, \tag{7.52}$$

subject to the constraints

$$\begin{aligned}
\sum_{(i,j)\in\mathcal{A}(i)} f_{ijrk} &= v_{irk}/d_{ik}, & i \in \mathcal{N}, \quad & (7.53)\\
\sum_{(i,j)\in\mathcal{A}(j)} f_{ijrk} &= v_{jrk}/d_{jk}, & j \in \mathcal{N}, \quad & (7.54)\\
f_{ijrk} + \alpha_{rk} &\leq \min\{v_{irk}/d_{ik}, v_{jrk}/d_{jk}\} z^*_{ijk}, & (i,j) \in \mathcal{A}. \quad & (7.55)
\end{aligned}$$

If $\alpha_{rk} > 0$ for any $k \in \mathcal{K}$ and $r \in \mathcal{R}$, then the current vector $\boldsymbol{z}^*$ is infeasible. The separation problem (7.52)–(7.55) is exactly the same as the RFSEP-DA separation problem (6.32)–(6.35), except for the additional scenario index $k \in \mathcal{K}$.

Theorem 7.3. *For a given $k \in \mathcal{K}$, $r \in \mathcal{R}$ and $\boldsymbol{z}^* \in \{0,1\}^{|\mathcal{A}|\times|\mathcal{K}|}$, the cut*

$$\sum_{(i,j)\in\mathcal{A}} \min\{v_{irk}/d_{ik}, v_{jrk}/d_{jk}\} z_{ijrk}\mu_{ij} + \sum_{i\in\mathcal{N}} v_{irk}/d_{ik}\left(\pi_i^1 + \pi_i^2\right) \leq 0, \tag{7.56}$$

where $\boldsymbol{\pi}^1 \in \mathbb{R}^{|\mathcal{N}|}$, $\boldsymbol{\pi}^2 \in \mathbb{R}^{|\mathcal{N}|}$ and $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{A}|}_{\leq 0}$ are the dual vectors associated with (7.53), (7.54) *and* (7.55)*, respectively, separates the infeasible point $\boldsymbol{z}^*$, if $\alpha_{rK} > 0$.*

Proof. The proof is analogous to the proof of Theorem 6.3. □

Valid inequalities

The explicit and implicit cutset inequalities proposed in Section 6.2.10 are easily extended to accommodate uncertainty in resource requirements, due to the assumption that the scenarios in the set $\mathcal{K}$ are independent of each other.

Proposition 7.3. *For any scenario $k \in \mathcal{K}$, resource $r \in \mathcal{R}$ and a subset $\mathcal{V} \subseteq \mathcal{N}$, the valid inequality*

$$\sum_{(i,j)\in\mathcal{A}(\mathcal{V},\bar{\mathcal{V}})} \min\{v_{irk}/d_{ik}, v_{jrk}/d_{jk}\} z_{ijk} \;\geq\; \max_{i\in\mathcal{V}}\{v_{irk}/d_{ik}\} \tag{7.57}$$

must hold in order for any given solution $\boldsymbol{z}^*$ to be feasible.

Proof. The proof is analogous to the proof of Theorem 6.1. □

Proposition 7.4. *For any scenario $k \in \mathcal{K}$ and for any subset $\mathcal{V} \subseteq \mathcal{N}$, the valid inequality*

$$\sum_{(i,j)\in\mathcal{A}(\mathcal{V},\bar{\mathcal{V}})} z_{ijk} \geq |\mathcal{V}| \tag{7.58}$$

must hold in order for any given solution $\boldsymbol{z}^*$ to be feasible.

Proof. The proof is analogous to the proof of Theorem 6.1. □

A maximum flow problem is solved for each scenario $k \in \mathcal{K}$, in order to identify a minimum cut. Let z^*_{ijk} be the LP relaxation solution to the RFM problem at a branch-and-bound node, for all $(i,j) \in \mathcal{A}$ and all $k \in \mathcal{K}$. By employing the preflow–push algorithm of Goldberg and Tarjan [40] for each $k \in \mathcal{K}$ where z^*_{ijk} are the arc capacities, a minimum cut $\mathcal{A}(\mathcal{V},\bar{\mathcal{V}})$ is obtained and used to derive an implicit cutset inequality (7.58).

Generating primal feasible solutions

The only modification required to the RSPH (introduced in Section 6.2.11) for generating primal feasible solutions within a branch-and-cut framework when solving the RF-STOCH-MAX problem, is to execute the RSPH algorithm for each scenario $k \in \mathcal{K}$. The scenarios are treated independently with the result that the RSPH determines a start time solution s^*_{ik} for each activity $i \in \mathcal{N}$ and for each scenario $k \in \mathcal{K}$. The solution vector $\boldsymbol{s}^*$ is used to derive the proposed sequencing order by setting $z^*_{ijk} = 1$ for all resource flow arcs $(i,j) \in \mathcal{A}$ and for all scenarios $k \in \mathcal{K}$, provided that $s^*_{jk} \geq s^*_{ik} + d_{ik}$. The solution vector $\boldsymbol{z}^*$ is subsequently used to fix the ordering variables $\boldsymbol{z}$ in a copy of the RFM problem. Solving this copy produces a primal solution to the RF-STOCH-MAX formulation.

7.3.3 Test data

Scenario-based data sets were generated based on the J30(50)DR×10, J60(50)DR×10, J90(50)DR×10, BLDR×10 and CNDR×10 data sets. For each instance in these data sets, random resource requirements were generated. Let v_{ir} denote the resource requirement of an activity $i \in \mathcal{N}$ for a resource $r \in \mathcal{R}$, based on an instance from J30(50)DR×10, J60(50)DR×10, J90(50)DR×10, BLDR×10 or CNDR×10. For each activity $i \in \mathcal{N}$ and resource $r \in \mathcal{R}$, a random vector $\delta_{ir} \in \mathbb{R}^{|\mathcal{K}|}$ is drawn from the normal distribution $N(0,1)$ in order to generate the new scenario-based resource requirements $v_{irk} = v_{ir}(1 + \delta_{irk})$. Care is taken to ensure feasibility by letting $\max\{0, v_{irk}\} \leq v_{irk} \leq \min\{U_r, v_{irk}\}$. The names of the newly created scenario-based data sets are augmented with a post-fix "ST."

In order to report on the tractability of the newly generated problem instances, the formulas for RS and DR (introduced in Section 5.1) are adapted to cater for scenario-based resource requirements. The resource strength of each resource $r \in \mathcal{R}$ is defined as

$$RS_r = \min_{k \in \mathcal{K}} \left\{ \frac{U_r - u_r^{\min}}{u_r^{\max} - u_r^{\min}} \right\},$$

where $u_r^{\min} = \max\{v_{ir} | i \in \mathcal{N}\}$ and $u_r^{\max}$ is the peak consumption of the resource $r \in \mathcal{R}$ over all the activities at any given point in time, when a feasible solution is constructed based only on the earliest start times of the activities (see Section 4.5.2). The resource strength of a problem instance is the average $RS = 1/|\mathcal{R}| \sum_{r \in \mathcal{R}} RS_r$. The disjunction ratio adapted for scenario-based resource requirements is defined as

$$DR = \frac{|TC(G(\mathcal{N}, \mathcal{Z}))| \cup \mathcal{D}}{(|\mathcal{N}|(|\mathcal{N}| - 1))/2},$$

where $TC(G(\mathcal{N}, \mathcal{Z}))$ is the transitive closure of the precedence graph and

$$\mathcal{D} = \{(i,j) \in \mathcal{A} | \ \exists \ k \in \mathcal{K}, r \in \mathcal{R}, v_{irk} + v_{jrk} > U_r\}$$

is the set of pairs of activities *in disjunction*.

	OS		NC		RF		RS		DR	
Data set	avg	σ	avg	σ	avg	σ	avg	σ	avg	σ
J30(50)DR×10ST×10	0.45	0.07	1.64	0.21	0.25	0.12	0.30	0.19	0.90	0.10
J60(50)DR×10ST×10	0.31	0.04	1.58	0.13	0.27	0.13	0.09	0.12	0.86	0.16
J90(50)DR×10ST×10	0.26	0.03	1.56	0.12	0.28	0.12	0.04	0.08	0.85	0.17
BLDR×10ST×10	0.34	0.07	2.93	0.46	0.25	0.03	0.3	0.09	0.9	0.06
CNDR×10ST×10	0.23	0.07	1.8	0.27	0.39	0.03	0.1	0.09	0.99	0.03

TABLE 7.5: *Tractability indicators for the randomly generated problem instances based on ten stochastic scenarios.*

For the purpose of evaluating the scalability of the Benders decomposition approach when uncertainty in resource requirements is considered, a set comprising ten scenarios was generated (that is, $|\mathcal{K}| = 10$). The newly generated data sets with scenario-based resource requirements are called the J30(50)DR×10ST×10, J60(50)DR×10ST×10, J90(50)DR×10ST×10, CNDR×10ST×10 and BLDR×10ST×10 data sets. Table 7.5 contains a summary of the statistics for these instances. Compared to the statistics of the base data sets, provided in Table 6.1, the DR values of the newly generated data sets with scenario-based resource requirements are larger, whereas the RS values are significantly smaller. Based only on the magnitude of the RS values, it is anticipated that the newly generated problem instances may pose more of a challenge to solve.

		Solved to optimality		Feasible Solutions	
Data set	Model	Instances (%)	Time (s)	Instances (%)	Gap (%)
J30(50)DR×10ST×10	RF	22	85.4	42	9.9
	RF-GR	8	113.2	70	19.8
	Benders	0	—	100	3.3
J60(50)DR×10ST×10	Benders	0	—	100	4.9
	RF-GR	0	—	66	283.4
	RF	0	—	6	17.5
J90(50)DR×10ST×10	Benders	0	—	100	6.1
	RF-GR	0	—	68	2041.5
	RF	0	—	0	—
BLDR×10ST×10	RF	2.6	225.4	100	135.6
	Benders	2.6	275.1	100	15.3
	RF-GR	0	—	100	85.6
CNDR×10ST×10	Benders	0	—	100	24.3
	RF	0	—	62.5	174.3
	RF-GR	0	—	50	97.7

TABLE 7.6: *Computational results obtained for the RCSP when taking uncertainty in the resource requirements into account.*

7.3.4 Computational results

The results obtained when solving the RCSP while taking into account uncertainty in the resource requirements, are listed in Table 7.6. As a first observation, the only data sets for which optimal solutions could be computed for some of the problem instances are the "easier" J30(50)DR×10ST×10 and BLDR×10ST×10 data sets. For all the other data sets, no optimal solutions could be computed for any of the problem instances.

The results do demonstrate, however, the superior scalability of the Benders decomposition approach. Feasible solutions were computed for all of the problem instances by employing Benders decomposition. These solutions were also of high quality as indicated by the low average integrality gaps obtained. For instance, the average integrality gap reported for the J60(50)DR×10ST×10 data set when applying Benders decomposition is 4.9%, compared to the 17.5% and 283% when solving the RCSP with RF-AC-GR and the plain RF formulation, respectively. For the J90(50)DR×10ST×10 data set, an average integrality gap of 6.1% is obtained by applying Benders decomposition, compared to 2 041% when applying the RF-AC-GR formulation. The plain RF formulation could not produce any feasible solutions for this data set.

7.4 Summary and conclusion

In this chapter, modelling extensions were proposed to cater for modelling requirements that are specific to an underground mining application. Algorithmic modifications were necessary to accommodate these new modelling requirements within a branch-and-cut framework.

Resource transfer delay constraints were incorporated into the resource flow formulation of the RCSP. These constraints are useful for accurately modelling crew and equipment movement in an underground mine. A small example was presented to illustrate the practical use of transfer delays and computational results were provided to demonstrate how the scalability of the Benders decomposition approach was maintained when taking into account transfer delays.

The resource flow formulation of the RCSP was also adapted to facilitate the selective scheduling of activities. The motivation for selective scheduling stems from the fact that some mining areas may be left behind (*i.e.* not excavated) and discarded permanently due to economic considerations. A small example was presented to support this argument and computational results were provided to demonstrate the efficiency of the Benders decomposition approach in maintaining scalability when taking selective scheduling into account.

A scenario-based approach was finally followed to capture uncertainty in the resource requirements of mining activities. A two-stage stochastic modelling approach was followed to ensure that the resulting schedule is both economically attractive and robust whenever future stochastic scenarios realise. Once again, a small example was presented to demonstrate the value of taking into account uncertainty in resource requirements when solving an underground mine scheduling optimisation problem. The superior scalability of the Benders decomposition approach was again reported through computational results that were based on randomly generated test data.

CHAPTER 8

Case Study

Contents

The computational results presented in earlier chapters, which were based on randomly generated problem instances, demonstrated the scalability of the proposed variable and constraint reduction approaches, as well as the proposed Benders decomposition of the RCSP. Although promising results were obtained, it remains to be seen whether these approaches are also successful in producing satisfactory results when considering real-world problem instances.

Mine planning data for a South African underground mine were used as a reference for constructing 12 real-world problem instances. The name of the mining company and financial-specific information may, however, not be disclosed due to the sensitive nature of the data.

8.1 Data specification

Recall from Chapter 3 that an underground mine is demarcated vertically into several levels. The number of levels are typically an indication of the size of a mining operation. One of the attributes associated with a mining activity is the level to which it belongs. Therefore, problem instances of varying sizes may be created by filtering the database of activities according to mine level identifiers. For the purpose of this study, the mine planning data at hand were partitioned into three smaller sets comprising six, ten and fourteen levels, respectively. By doing this, real-world problem instances corresponding to a small operation with six levels, a medium operation with ten levels and a large operation with fourteen levels, are considered. Some level of aggregation was performed on activities with similar characteristics. More specifically, individual stoping panels along a raise line were aggregate to form a single stoping block. This reduced the final number of activities within each data set.

In addition to filtering activities based on mine level identifiers, a planning horizon criterion was applied in the final construction of the different problem instances. By making use of earliest start times of the activities (computed as part of the preprocessing procedure discussed in Section 4.5.2), problem instances corresponding to specific planning horizons were created. More

Instance name	Number of levels	Planning horizon	Number of Activities
L6-Y2	6	2 years	53
L6-Y5	6	5 years	106
L6-Y10	6	10 years	212
L6-Y15	6	15 years	368
L10-Y2	10	2 years	109
L10-Y5	10	5 years	179
L10-Y10	10	10 years	369
L10-Y15	10	15 years	539
L14-Y2	14	2 years	163
L14-Y5	14	5 years	269
L14-Y10	14	10 years	540
L14-Y15	14	15 years	766

TABLE 8.1: *Problem instances created from real mine planning data.*

specifically, problem instances were created for a 2-year, 5-year, 10-year and 15-year planning horizon, respectively.

Table 8.1 provides information on the problem instances created for varying mining operation sizes and different planning horizons. The last column of the table contains the numbers of activities included in each of the problem instances.

Computational results are provided in the remainder of this chapter to showcase the efficiency of the proposed problem formulations and algorithmic approaches when applied to the problem instances listed in Table 8.1. It is, however, informative to consider the practical aspects in underground mine scheduling and how they are accommodated in the mathematical models considered in this dissertation.

8.2 Practical considerations

Recall the small mine scheduling example presented in Section 3.2. As indicated by the input data for this example in Table 3.1, resource consumption/production values v_{ir} for each resource $r \in \mathcal{R}$ are associated with each activity $i \in \mathcal{N}$. The same approach applies to the problem instances considered in this section. The main resources of concern in this section are related to infrastructure capacities and crew requirements.

Each level of a typical underground mine is demarcated into two sections, with each section referred to as a *half-level*. Capacity constraints related to the volume of rock that may be trammed within a given level are specified per half-level. A typical half-level capacity is a value of between 10 000 to 20 000 tonnes per month, depending on the layout and the physical infrastructure of the mine.

For illustrative purposes, the L6-Y2 problem instance is considered and the resources that are reported on include tonnes per half-level and the number of crews required for stoping activities. In practice there may be several other resources required to capture all of the costing parameters and resource limitations. The motivation for the development of algorithmic approaches to improve computing times of RCSP problem instances with large numbers of resources now becomes apparent, considering that for the L6-Y2 problem instance, a minimum of twelve resources are required, one for each half-level.

As a first course of action typically followed in practice, an unconstrained version of the L6-Y2 problem instance is solved. The tonnes profile for the twelve half-levels obtained by solving the

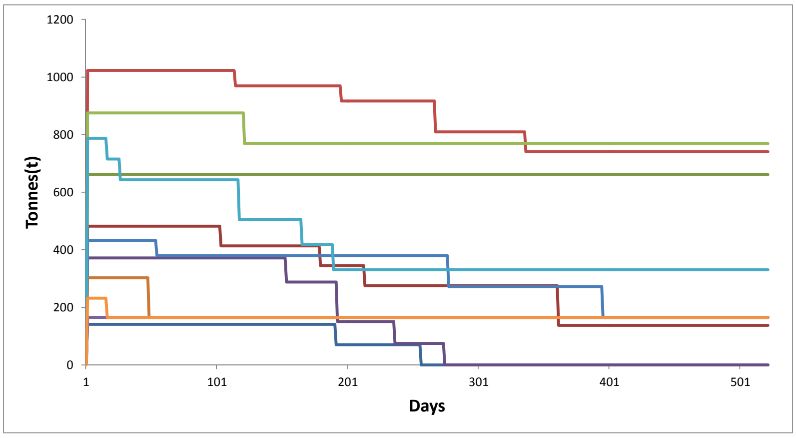


FIGURE 8.1: *Unconstrained tonnes profile for the twelve half-levels of the L6-Y2 problem instance.*

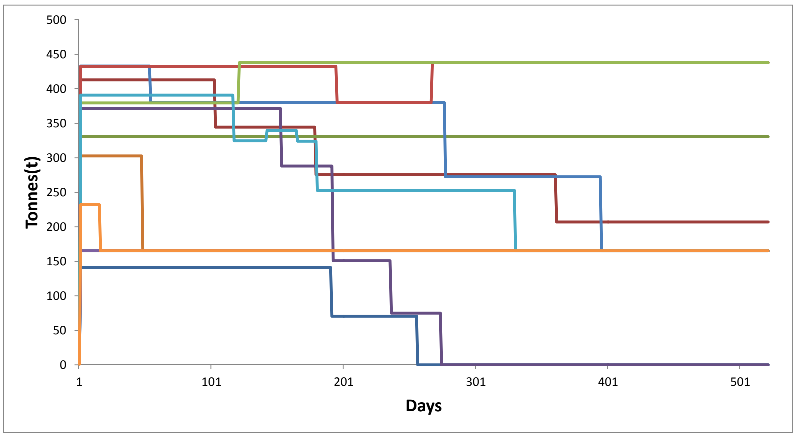


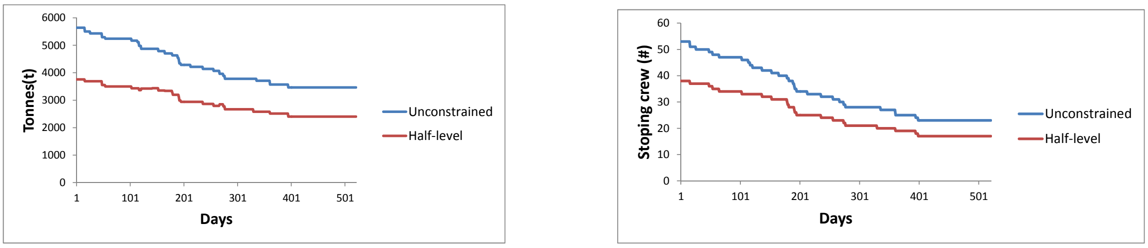
FIGURE 8.2: *Tonnes profile for the twelve half-levels of the L6-Y2 problem instance when considering a half-level capacity constraint of 10 000 per month.*

RF-MAX problem formulation without any resource restrictions, is depicted in Figure 8.1. Note that the resulting solution is a schedule based on a daily calendar. Therefore, the tonnes profile in Figure 8.1 is per day.

An unconstrained scheduling solution is typically required in order to validate the precedence requirements of the activities, specified as sequencing rules. By considering a tonnes profile, an experienced mine planner should be able to assess quickly whether erroneous input data were used in the construction of the scheduling solution. Furthermore, the starting solutions of the activities may be imported into a 3D mine planning system and the resulting graphical display of the scheduling solution may be used for further validation.

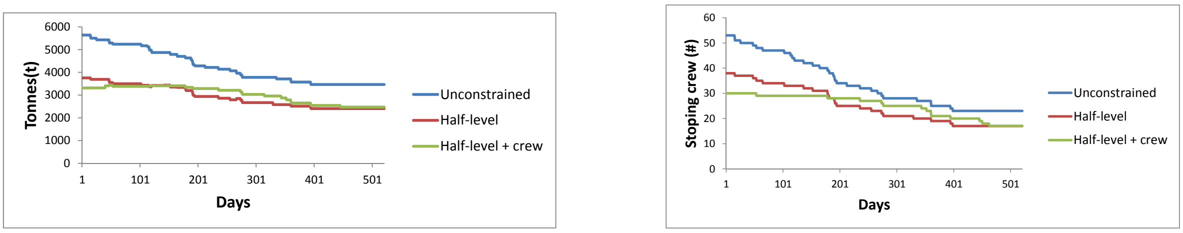
The next step, following the validation of the unconstrained schedule, is the systematic application of the remaining constraints. For instance, a half-level capacity of 10 000 tonnes per month is assumed for all of the remaining empirical tests that involve the L6, L10 and L14 problem instances. The tonnes profile obtained when re-solving the L6-Y2 problem instance while taking the half-level capacity constraint into account is given in Figure 8.2. Note that since the resulting tonnes profile is displayed per day, the maximum allowable tonnes per day is given by $10\,000 \times \frac{12}{260} \approx 462$, if the number of production days in a year is assumed to be 260.

The tonnes profiles depicted in Figure 8.3(a) are for the unconstrained case and for the case where the half-level capacity constraint is imposed. By applying the capacity constraint, a reduction in total tonnes can clearly be observed. The effect of the constraint is also noticeable



(a) *Total tonnes profile* (b) *Stoping crew requirements.*

FIGURE 8.3: *Comparing the effect on total tonnes and crew for the L6-Y2 problem instance when considering a half-level capacity of 10 000 tonnes per month.*



(a) *Total tonnes profile* (b) *Stoping crew requirements.*

FIGURE 8.4: *Comparing the effect on total tonnes and crew for the L6-Y2 problem instance when considering a half-level capacity of 10 000 tonnes per month and a maximum crew requirement limit of 30 crews per day.*

when considering the profiles of other resources. For instance, the crew requirement profile in Figure 8.3(b) clearly shows a reduction in the number of crews required over time, as a result of activities being scheduled to start at a later date in order to satisfy the capacity constraint.

As a further illustration of the effect that constraints have on resource profiles, consider the case where a limit is imposed on the number of crews available per day. The crew profile in Figure 8.3(b), for the case where the half-level capacity constraint is applied, suggests that a maximum of up to 38 crews are required to realise the corresponding tonnes profile. In the remainder of this chapter, however, a maximum limit of 30 crews per day is considered for the L6 problem instances, 40 crews per day for the L10 problem instances and 50 crews per day for the L14 problem instances.

The resulting profiles for the L6-Y2 problem instance, when considering a crew limit of 30 per day, are shown in Figure 8.4. The combined effect of the crew requirement and the half-level capacity constraints is clearly visible in the tonnes profile depicted in Figure 8.4(a). It is interesting to note, however, that although the volume of tonnes decreases during the early periods due to the crew requirement constraint, the tonnes profile is higher for subsequent days as a result of activities being scheduled to start later. This effect is also visible for the crew requirement profile shown in Figure 8.4(b).

8.3 Computational results

Results in this section were generated by considering the plain RF formulation, the RF formulation in conjunction with aggregated constraints and graph reduction (RF-AC-GR), and Benders decomposition. For the separation of feasibility cuts and the generation of primal feasible solutions, RFSEP-PDA and RSPH were applied, respectively.

The computational results that follow are for all of the problem instances listed in Table 8.1.

	RF		RF-AC-GR		Benders	
Problem instance	Time(s)	Gap(%)	Time(s)	Gap(%)	Time(s)	Gap(%)
L6-Y2	3	1.0%	5	1.0%	142	1.0%
L10-Y2	900	1.0%	790	1.0%	900	8.7%
L14-Y2	537	1.0%	81	1.0%	900	5.3%
L6-Y5	1 800	1.4%	1 800	1.3%	1 800	6.6%
L10-Y5	1 800	1.7%	1 800	1.1%	1 800	11.7%
L14-Y5	1 800	1.4%	1 800	1.0%	1 800	11.0%
L6-Y10	—	—	2 700	2.1%	2 700	18.2%
L10-Y10	—	—	2 700	1.9%	2 700	15.6%
L14-Y10	—	—	—	—	2 700	16.1%
L6-Y15	—	—	—	—	3 600	17.7%
L10-Y15	—	—	—	—	3 600	17.1%
L14-Y15	—	—	—	—	3 600	18.2%

TABLE 8.2: *The computational efficiency of solving real-world instances when considering a half-level capacity of 10 000 tonnes per month.*

	RF		RF-AC-GR		Benders	
Problem instance	Time(s)	Gap(%)	Time(s)	Gap(%)	Time(s)	Gap(%)
L6-Y2	36	1.0%	6	1.0%	900	1.0%
L10-Y2	900	3.3%	900	1.4%	900	8.4%
L14-Y2	900	11.8%	900	8.6%	900	5.4%
L6-Y5	1 800	2.8%	1 800	1.9%	1 800	9.7%
L10-Y5	1 800	4.9%	1 800	14.1%	1 800	11.8%
L14-Y5	—	—	1 800	9.2%	1 800	11.0%
L6-Y10	—	—	2 700	10.1%	2 700	19.3%
L10-Y10	—	—	2 700	14.3%	2 700	16.7%
L14-Y10	—	—	—	—	2 700	17.1%
L6-Y15	—	—	—	—	3 600	18.9%
L10-Y15	—	—	—	—	3 600	18.4%
L14-Y15	—	—	—	—	3 600	19.6%

TABLE 8.3: *The computational efficiency of solving real-world instances when considering a half-level capacity of 10 000 tonnes per month and a maximum crew requirement limit of 30 crews per day.*

A time limit of 900 seconds was imposed on solution times for Y2 problem instances, a limit of 1 800 seconds for the Y5 problem instances, 2 700 seconds for the Y10 problem instances and 3 600 seconds for the Y15 problem instances.

The results obtained when solving all of the problem instances by taking the half-level capacity constraints into account, are provided in Table 8.2. Positive results are reported for problem instances relating to the two-year and five-year planning horizons. Optimal solutions were computed for L6-Y2 and L14-Y2 by applying both the RF and RF-AC-GR formulations. The corresponding solution times were 3 seconds and 537 seconds, respectively. The L6-Y2 problem instance was also solved to optimality by means of the RF-AC-GR formulation as as well as by employing Benders decomposition. The corresponding solution times were 5 seconds and 142 seconds, respectively. Very small integrality gaps were measured for all the other instances not solved to optimality, but for which at least one feasible solution could be computed. The scalability of the Benders decomposition approach is demonstrated through its ability to compute feasible solutions to all of the problem instances listed in Table 8.2.

Table 8.3 shows the computational results obtained when taking both the half-level capacity constraints and the crew requirements constraint into account. As a first observation it is evident that integrality gaps are much larger for both the RF and the RF-AC-GR, and that optimal

	RF		RF-AC-GR		Benders	
Problem instance	Time(s)	Gap(%)	Time(s)	Gap(%)	Time(s)	Gap(%)
L6-Y2	17	1.0%	55	1.0%	900	2.0%
L10-Y2	900	4.7%	900	2.2%	900	8.3%
L14-Y2	900	18.5%	900	109.5%	900	5.5%
L6-Y5	1 800	5.3%	1 800	3.3%	1 800	10.5%
L10-Y5	1 800	18.7%	1 800	150.1%	1 800	12.3%
L14-Y5	1 800	201.2%	1 800	∞%	1 800	18.8%
L6-Y10	2 700	62.6%	2 700	89.9%	2 700	18.9%
L10-Y10	2 700	542.3%	2 700	∞%	2 700	23.3%
L14-Y10	—	—	—	—	2 700	23.9%
L6-Y15	—	—	3 600	526.0%	3 600	18.8%
L10-Y15	—	—	—	—	3 600	25.6%
L14-Y15	—	—	—	—	—	—

TABLE 8.4: *The computational efficiency of solving real-world instances when considering a half-level capacity of 10 000 tonnes per month, a maximum crew requirement limit of 30 crews per day and the option to allow selective scheduling.*

solutions could be computed for only the L6-Y2 problem instance. Although it was not possible to produce any optimal solutions with Benders decomposition, it is still very encouraging to observe that there was a marginal increase in integrality gaps when solving the problem instances with additional constraints. This is once again a confirmation of the superior scalability achieved in solving RCSP problem instances using Benders decomposition.

Due to the unavailability of data, no empirical tests could be performed as part of the case study to evaluate the computational efficiency when considering transfer delays and uncertainty in resource requirements. Computational results are, however, available for investigating the effect on computing times when applying selective scheduling on the real-world problem instances, since no additional data are required. These results are provided in Table 8.4. It is interesting to note, compared to the previous results, that more feasible solutions could be computed using the plain RF formulation and the RF-AC-GR formulation. For instance, the results in Table 8.3 show that feasible solutions could be computed only up to problem instance L10-Y5 by employing the plain RF formulation, whereas in Table 8.4, it is shown that feasible solutions were computed for problem instances up to L10-Y10. The L6-Y15 problem instance is the only additional problem instance for which feasible solutions could be generated by employing the RF-AC-GR formulation. It should be noted, however, that although more feasible solutions could be computed, the quality of these solutions are poor and in some cases correspond to solutions where no activities have been scheduled. More specifically, these solutions dictate that it is economically more viable to do no excavation at all, resulting in an NPV of zero and, consequently, in an infinite integrality gap.

The results obtained when applying Benders decomposition are still very promising in the context of selective scheduling. The results in Table 8.4 demonstrate, once again, the scalability achieved in solving RCSP problem instances using Benders decomposition.

8.4 Summary

The results presented in this chapter are based on problem instances which were created from real mine planning data. Twelve instances were created by considering different combinations of the number of mining levels and planning horizons. The practical implication of applying real-world capacity and crew constraints were demonstrated by considering one of these problem instances.

The plain RF formulation failed to provide any feasible solutions to problem instances that correspond to planning horizons extending beyond a five-year period when considering capacity and crew constraints. Feasible solutions were computed via the plain RF formulation for two larger problem instances, but only when allowing selective scheduling. The resulting solutions were, however, of poor quality.

Some improvement was possible by employing the RF-AC-GR problem formulation, but still no feasible solutions could be computed for problem instances that correspond to planning horizons extending beyond a ten-year period and for which capacity and crew constraints were applied. By allowing selective scheduling, feasible solutions could be computed for one problem instance having a fifteen-year planning horizon.

The use of Benders decomposition resulted in the computation of feasible solutions for all of the problem instances. The only exception was when selective scheduling was allowed, in which case no feasible solutions could be computed for the largest of the twelve problem instances.

CHAPTER 9

Summary and conclusion

Contents

The global economic downturn in recent years and the increasing competitiveness of corporations as a result of globalisation are only some of the challenges faced by mining companies. In order to remain profitable, proper planning and sound decision making are imperative for managing the complexity of mining operations.

A key aspect in the production planning life cycle at an underground mine is the use of highly sophisticated mine planning tools to assist with decision making. The latest 3D CAD planning systems encapsulate both mine layout design functionality and computer algorithms for generating production schedules. Although significant progress has been made over the last couple of decades in developing improved mine planning systems, the ability of these systems to cater for all of the mining complexities is still very limited. Furthermore, as computing abilities improve, the demand for more functionality in these systems increases.

The use of mathematical models and algorithms play an important role in the development of mine planning systems. More specifically, it was shown in this dissertation that the underground mine scheduling problem is a special case of the well-known RCSP. The approach followed in this study has therefore been to obtain an in-depth understanding of the theory underlying the RCSP, and to explore algorithmic approaches that show potential in terms of improving computing efficiency when solving an underground mine scheduling problem.

A summary of the content of this dissertation is provided in the first section of this chapter. The findings of this study and explicit reference to contributions made in the dissertation are summarised in the following section. The chapter closes with a brief overview of possible future follow-up work and a discussion on potentially open research questions related to the topic of this dissertation.

9.1 Chapter summaries

The introductory chapter of this dissertation provided context for the motivation to embark on this study. The use of appropriate mathematical models and algorithms has become a necessity for mining companies to ensure that short-term and long-term production plans culminate in sustainable profits. The primary goal of this study was to improve on existing mathematical models in order to address as much of the typical underground mine planning requirements as possible. Furthermore, since the solution of these models is inherently hard, the search for algorithmic enhancements was identified as the secondary objective of this study.

An overview of the theoretical foundations of scheduling algorithms was provided in Chapter 2. The emphasis in this chapter was on the formulation of resource constrained scheduling problems and algorithmic approaches towards providing solutions to these problems. A very basic introduction to complexity theory provided a framework for discussing the hardness of problem instances and ways of measuring the efficiency of algorithms. A literature review was also presented of the most recent contributions to the field of resource constrained scheduling. The scope of the literature study was, however, limited to work within an exact solution framework.

Chapter 3 provided an overview of the technical aspects pertaining to underground mining operations and the process of mine planning and scheduling. It provided a glimpse into the complexities involved in planning the execution of mining activities while taking infrastructure and resource limitations into account. An important contribution of this chapter was the use of a small example to demonstrate that the underground mine scheduling optimisation problem is a special case of the well-known resource constrained scheduling problem. Furthermore, it was also shown that the standard framework for handling resource constrained scheduling problems does not necessarily accommodate all of the operational requirements encountered in underground mining. It is for this reason that an exact modelling framework approach was adopted in this dissertation, in order to address some of these requirements.

Three main classes of RCSP formulations were presented in Chapter 4, namely *time-indexed formulations*, *resource flow-based formulations*, and *event-based formulations*. The objective of this chapter was to examine the computational properties of the these formulations and to propose new mathematical constructs that facilitate the maximisation of NPV in both the resource flow-based and event-based formulations. In addition, a preprocessing approach was described which improves the bounds on some of the variables for the three RCSP formulations and which may be used for the generation of initial feasible solutions.

Chapter 5 contained a presentation of empirical results on the computational properties of the above RCSP formulations. Randomly generated problem instances were introduced, which are based on well-known instances from the literature and which were generated so that they better reflect characteristics of a typical underground mine scheduling optimisation problem. Another important contribution of this chapter was the characterisation of the newly proposed problem instances according to various tractability indicators. The results reported in this chapter showed that the resource flow-based formulation is computationally more efficient compared to the time-indexed and the event-based formulations. This is especially true for the newly created problem instances having longer activity durations. Based on these findings, subsequent algorithmic enhancements and model extensions were all based on the resource flow-based formulation.

The heart of this dissertation is captured in the contents of Chapters 6 and 7. In Chapter 6, reformulations of the resource flow-based RCSP were proposed for the purpose of improving computing times when solving problem instances that are typical of an underground mine. As a first suggestion, a constraint aggregation approach was formulated which resulted in a reduced number of resource flow-related constraints. A graph reduction approach was also suggested for reducing the number of variables in the resource flow-based formulation, by taking the logical movement of resources into account with respect to the underlying precedence graph. Details of a Benders decomposition approach were provided, and theorems were established in support of the derivation of four separation sub-problems. The implementation of these separation routines as part of a branch-and-cut approach was described and a heuristic approach was suggested for the purpose of generating primal solutions. The chapter closed with an innovative suggestion of applying valid inequalities that resemble the well-known cutset inequalities found in solution approaches for solving network design problems.

The success of the proposed model reformulations and algorithmic enhancements were showcased in Chapter 6 as part of an empirical study. An extensive report was provided detailing computational results for randomly generated data sets, which resembles problem instances typically found in underground mining. The main conclusion drawn from the empirical work performed is that most of the proposed enhancements improve computing times when compared to the application of existing formulations found in the literature. The Benders decomposition approach, implemented within a branch-and-cut framework, was shown to scale well for problem instances with a large number of activities and resources. This is a significant contribution within the context of mining, especially considering the large number of resources that have to be accommodated when solving underground mine scheduling optimisation problems.

The objective of Chapter 7 was to present modelling extensions of the resource flow-based RCSP in order to address mining-specific modelling requirements. Of specific interest was the introduction of transfer delays, the facilitation of selective mining and, finally, the modelling of uncertainty in the resource requirements of the RCSP. In addition to the modifications required to the resource flow-based problem formulation, algorithmic changes were required to maintain the correct implementation of the heuristics and separation routines within the branch-and-cut framework. The computational results reported in Chapter 7 demonstrated the efficiency of the Benders decomposition approach in maintaining scalability while taking the various mining-specific modelling requirements into account.

Chapter 8 provided final proof of the quality of the algorithmic contributions presented in this dissertation. The scalability of the proposed variable and constraint reduction approaches, as well as the proposed Benders decomposition of the RCSP, was demonstrated in the context of real-world mine planning problem instances. More specifically, the results reported in Chapter 8 showed that the computation of feasible solutions was made possible by Benders decomposition, even for problem instances for which no feasible solutions could be computed by employing the original resource flow-based RCSP formulation.

9.2 Future work

Computational results obtained for the branch-and-cut approach demonstrated the success of applying a decomposition approach. The inability to compute feasible solutions for some of the real-world instances is, however, enough reason to further efforts towards developing more efficient approaches. The establishment of the branch-and-cut framework for solving the resource constraint scheduling problem also opened up the opportunity to address other modelling requirements, which were previously unattainable due to the inefficiency of existing solution approaches. Details of possible further algorithmic work and modelling extensions are provided below:

1. **Precedence-related valid inequalities**. The order in which activities may be scheduled is determined by the precedence relationships which are captured in the precedence graph. A possible extension to the work in this dissertation is the exploration of information locked up in the precedence graph in order to allow for the derivation of valid inequalities. Once again these inequalities may be added explicitly as part of the problem formulation or generated during the branch-and-bound process by means of a separation routine. The work by Hardin *et al.* [42] is suggested as a basis for further work on this topic.

2. **Column generation for the resource flow variables**. The Benders decomposition approach proposed in this dissertation involves a separation sub-problem that contains variables related to the flow of resources. The number of resource flow variables grows exponentially with an increase in the number of activities. The inability to compute solutions to large-scale problems may, therefore, be attributed to insufficient memory as a result of the enormous growth in variables. The use of column generation approaches within a branch-and-cut framework has been applied successfully in network design applications for improving memory usage [76]. It should, therefore, be possible to incorporate a column generation approach within the branch-and-cut framework suggested in this dissertation for generating the required resource flow-related variables during the branch-and-bound process.

3. **Constraint programming as primal heuristic**. The application of constraint programming has demonstrated to be very efficient in generating feasible solutions, and in several cases optimal solutions, to the resource constraint scheduling problem [63]. It is, therefore, reasonable to expect that the application of constraint programming within a branch-and-cut approach may improve computing times when applied as a primal heuristic.

References

[1] Alagöz O & Azizoglu M, 2003, *Rescheduling of identical parallel machines under machine eligibility constraints*, European Journal of Operational Research, **149(3)**, pp. 523–532.

[2] Artigues C, Demassey S & Nèron E (Eds), 2008, *Resource-constrained project scheduling: Models, algorithms, extensions and applications*, John Wiley & Sons Inc., Hoboken (NJ).

[3] Artigues C, Leus R & Nobibon FT, 2013, *Robust optimization for resource-constrained project scheduling with uncertain activity durations*, Flexible Services and Manufacturing Journal, **25(1)**, pp. 175–205.

[4] Artigues C, Michelon P & Reusser S, 2003, *Insertion techniques for static and dynamic resource-constrained project scheduling*, European Journal of Operational Research, **149(2)**, pp. 249–267.

[5] Artigues C & Roubellat F, 2000, *A polynomial insertion algorithm in a multi-resource schedule with cummulative constraints and multiple modes*, European Journal of Operational Research, **127(2)**, pp. 297–316.

[6] Ashtiani B, Leus R & Aryanezhad M, 2009, *New competitive results for the stochastic resource-constrained project scheduling problem: the benefits of pre-processing*, Journal of Scheduling, **14(2)**, pp. 157–171.

[7] Askari-Nasab H, Pourrahimian Y, Ben-Awuah E & Kalantari S, 2011, *Mixed integer linear programming formulations for open pit production scheduling*, Journal of Mining Science, **47(3)**, pp. 338–359.

[8] Atamtürk A, 2002, *On capacitated network design cut–set polyhedra*, Mathematical Programming, **92(3)**, pp. 425–437.

[9] Ayala M, Artigues C & Gacias B, 2010, *Lagrangian relaxation-based lower bound for resource-constrained modulo scheduling*, Electronic Notes in Discrete Mathematics, **36**, pp. 191–198.

[10] Baptiste P & Demassey S, 2004, *Tight LP bounds for resource-constrained project scheduling*, OR Spectrum, **26(2)**, pp. 251–262.

[11] Baptiste P & Pape CL, 2000, *Constraint propagation and decomposition techniques for highly disjunctive and highly cumulative project scheduling problems*, Constraints, **5(1)**, pp. 119–139.

[12] Benders J, 1962, *Partitioning procedures for solving mixed-variables programming problems*, Numerische Mathematik, **4**, pp. 238–252.

[13] Bienstock D & Günlük O, 1994, *Capacitated network design — polyhedral structure and computation*, INFORMS Journal on Computing, **8**, pp. 243–259.

[14] Blazewicz J, Lenstra J & Rinnooy Kan A, 1983, *Scheduling subject to resource constraints: Classification and complexity*, Discrete Applied Mathematics, **5(1)**, pp. 11–24.

[15] Bley A, Boland N, Fricke C & Froyland G, 2010, *A strengthened formulation and cutting planes for the open pit mine production scheduling problem*, Computers and Operations Research, **37(9)**, pp. 1641–1647.

[16] Brucker P, 2006, *Scheduling algorithms*, Springer Verlag, Berlin.

[17] Brucker P & Knust S, 2000, *A linear programming and constraint propagation-based lower bound for the RCPSP*, European Journal of Operational Research, **127**, pp. 355–362.

[18] Brucker P, Knust S, Schoo A & Thiele O, 1998, *A branch and bound algorithm for the resource-constrained scheduling problem*, European Journal of Operational Research, **107**, pp. 272–288.

[19] Caccetta L & Hill S, 2003, *An application of branch and cut to open pit mine scheduling*, Journal of Global Optimization, **27(2)**, pp. 349–365.

[20] Carlier J & Néron E, 2003, *On linear lower bounds for the resource constrained project scheduling problem*, European Journal of Operational Research, **149(2)**, pp. 314–324.

[21] Carlyle M & Eaves B, 2001, *Underground planning at Stillwater Mining Company*, Interfaces, **31(4)**, pp. 50–60.

[22] Chamber of Mines of South Africa, 2011, *Factsheet 2011*, url: `http://chamberofmines.org.za/media-room/facts-and-figures`.

[23] Chamber of Mines of South Africa, 2015, *Annual report for 2015*, url: `http://www.chamberofmines.org.za/industry-news/publications/annual-reports`.

[24] Christofides N, Alvarez-Valdés R & Tamarit J, 1987, *Project scheduling with resource constraints: A branch-and-bound approach*, European Journal of Operational Research, **29(3)**, pp. 262–273.

[25] Cook S, 1971, *The complexity of theorem-proving procedures*, Proceedings of the Third Annual ACM Symposium on Theory of Computing, Shaker Heights (OH), pp. 151–158.

[26] Cornuéjols G, 2008, *Valid inequalities for mixed integer linear programs*, Mathematical Programming, **112(1)**, pp. 3–44.

[27] Dantzig G, 1960, *On the significance of solving linear programming problems with some integer variables*, Econometrica, **28**, pp. 30–44.

[28] de Vonder SV, Demeulemeester E & Herroelen W, 2008, *Proactive heuristic procedures for robust project scheduling: An experimental analysis*, European Journal of Operational Research, **189(3)**, pp. 723–733.

[29] Deblaere F, Demeulemeester E, Herroelen W & de Vonder SV, 2007, *Robust resource allocation decisions in resource-constrained projects*, Decision Sciences, **38(1)**, pp. 5–37.

[30] Demassey S, Artigues C & Michelon P, 2005, *Constraint-propagation-based cutting planes: An application to the resource-constrained project scheduling problem*, INFORMS Journal on Computing, **17(1)**, pp. 52–65.

[31] Demeulemeester E & Herrolen W, 1992, *A branch-and-bound procedure for the multiple-resource constrained single project scheduling problem*, Management Science, **38(12)**, pp. 1803–1818.

[32] Doersch R & Patterson J, 1977, *Scheduling a project to maximize its present value: A zero-one programming approach*, Management Science, **23(8)**, pp. 882–889.

[33] Dorndorf U, Pesch E & Phan-Huy T, 2000, *A branch-and-bound algorithm for the resource-constrained project scheduling problem*, Mathematical Methods of Operations Research, **52**, pp. 413–439.

[34] Dorndorf U, Pesch E & Phan-Huy T, 2000, *Constraint propagation techniques for the disjunctive scheduling problem*, Artificial Intelligence, **122(1)**, pp. 189–240.

[35] Ecmeli O & Erenguc S, 1996, *A branch and bound procedure for the resource constrained project scheduling problem with discounted cash flows*, Manangement Science, **42(10)**, pp. 1395–1408.

[36] Epstein R, Goic M, Weintraub A, Catalán J, Santibáñez P, Urrutia R, Cancino R, Gaete S, Aguayo A & Caro F, 2012, *Optimizing long-term production plans in underground and open-pit copper mines*, Operations Research, **60(1)**, pp. 4–17.

[37] Francisco B, 1996, *Network design using cut inequalities*, SIAM Journal on Optimization, **6(3)**, pp. 823–837.

[38] Garey M & Johnson D, 1976, *Complexity results for multiprocessor scheduling under resource constraints*, SIAM Journal on Computing, **4(4)**, pp. 397–411.

[39] Gersmann K & Hammer B, 2005, *Improving iterative repair strategies for scheduling with the SVM*, Neurocomputing, **63**, pp. 271–292.

[40] Goldberg A & Tarjan R, 1986, *A new approach to the maximum flow problem*, Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing, New York (NY), pp. 136–146.

[41] Grahaman R, Lawler E, Lenstra J & Rinnooy Kan A, 1979, *Optimization and approximation in deterministic sequencing and scheduling: A survey*, Annals of Discrete Mathematics, **5**, pp. 287–326.

[42] Hardin R, Nemhauser G & Savelsbergh M, 2008, *Strong valid inequalities for the resource-constrained scheduling problem with uniform resource requirements*, Discrete Optimization, **5(1)**, pp. 19–35.

[43] Hartmann S & Briskorn D, 2010, *A survey of vairants and extensions of the resource-constrained project scheduling problem*, European Journal of Operational Research, **207**, pp. 1–14.

[44] Herroelen W & Leus R, 2004, *The construction of stable project baseline schedules*, European Journal of Operational Research, **156(3)**, pp. 550–565.

[45] Higle JL, 2005, *Stochastic programming: optimization when uncertainty matters*, pp. 30–53 in Smith C (Ed), *Emerging Theory, Methods, and Applications*, INFORMS, Hanover (MD).

[46] Horbach A, 2010, *A Boolean satisfiability approach to the resource-constrained project scheduling problem*, Annals of Operations Research, **181(1)**, pp. 89–107.

[47] Hustrulid W & Bullock R, 2001, *Underground mining methods: Engineering fundamentals and international case studies*, Society for Mining Metallurgy and Exploration, Littleton (CO).

[48] *IBM ILOG CPLEX*, IBM, New York, URL: `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`.

[49] Johnson T, 1967, *An algorithm for the resource constrained project scheduling problem*, PhD Thesis, Massachusets Institute of Technology, Cambridge (MA).

[50] Kazakidis V & Scoble M, 2003, *Planning for flexibility in underground mine production systems*, Mining Engineering, **55(8)**, pp. 33–39.

[51] Kelley J & Walker M, 1959, *Critical-path planning and scheduling*, Proceedings of the IRE-AIEE-ACM Computer Conference, New York (NY).

[52] Kolisch R & Padman R, 2001, *An integrated survey of deterministic project scheduling*, Omega, **29**, pp. 249–272.

[53] Kolish R & Sprecher A, 1996, *PSPLIB — A project scheduling problem library*, European Journal of Operational Research, **96**, pp. 205–216.

[54] Koné O, Artigues C, Lopez P & Mongeau M, 2011, *Event-based MILP models for resource-constrained project scheduling problems*, Computers and Operations Research, **38(1)**, pp. 3–13.

[55] Krige D, 1951, *A statistical approach to some basic mine valuation problems on the Witwatersrand*, Journal of the Chemical, Metallurgical and Mining Society of South Africa, **52(6)**, pp. 119–139.

[56] Krüger D & Scholl A, 2010, *Managing and modelling general resource transfers in (multi-)project scheduling*, OR Spectrum, **32(2)**, pp. 369–394.

[57] Laborie P, 2005, *Complete MCS-based search: Application to resource-constrained project scheduling*, Proceedings of the International Joint Conference on Artificial Intelligence, Edinburgh.

[58] Lambrechts O, Demeulemeester E & Herroelen W, 2007, *Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities*, Journal of Scheduling, **11(2)**, pp. 121–136.

[59] Land A & Doig A, 1960, *An automatic method of solving discrete programming problems*, Econometrica, **28(3)**, pp. 497–520.

[60] Lenstra J & Rinnooy Kan A, 1978, *Complexity of scheduling under precedence constraints*, Operations Research, **26(1)**, pp. 22–35.

[61] Leus R & Herroelen W, 2004, *Stability and resource allocation in project planning*, IIE Transactions, **36(7)**, pp. 1–16.

[62] Li H & Womer N, 2015, *Solving stochastic resource-constrained project scheduling problems by closed-loop approximate dynamic programming*, European Journal of Operational Research, **246(1)**, pp. 20–33.

[63] Liess O & Michelon P, 2008, *A constraint programming approach for the resource-constrained project scheduling problem*, Annals of Operations Research, **157(1)**, pp. 25–36.

[64] Linderoth J & Savelsbergh M, 1999, *A computational study of search strategies for mixed integer programming*, INFORMS Journal on Computing, **11**, pp. 173–187.

[65] Little J, Nehring M & Topal E, 2008, *A new mixed integer programming model for mine production scheduling optimisation in sublevel stope mining*, Proceedings of the Australian Mining Technology Conference, pp. 157–172.

[66] Markowitz H & Manne A, 1957, *On the solution of discrete programming problems*, Econometrica, **25**, pp. 84–110.

[67] Martinez M & Newman A, 2011, *A solution approach for optimizing long- and short-term production scheduling at LKAB's Kiruna mine*, European Journal of Operational Research, **211(1)**, pp. 184–197.

[68] *Mine2-4D*, MineRP, Johannesburg, URL: http://www.minerp.com/en/software-products/planningsched/Mine2-4D.aspx.

[69] Mingozzi A, Maniezzo V, Ricciardelli S & Bianco L, 1998, *An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation*, Management Science, **44(5)**, pp. 714–729.

[70] Mining Weekly, 2011, *South African mining industry committed to mechanisation*, URL: http://www.miningweekly.com/article/sa-mining-industry-committed-to-mechanisation-2015-04-03.

[71] Möhring R, Schulz A, Stork F & Uetz M, 2003, *Solving project scheduling problems by minimum cut computations*, Management Science, **49(3)**, pp. 330–350.

[72] Moodie C & Mandeville D, 1966, *Project resource balancing by assembly line balancing techniques*, Journal of Industrial Engineering, **17(7)**, pp. 377–383.

[73] Nehring M, Topal E & Little J, 2010, *A new mathematical programming model for production schedule optimization in underground mining operations*, The Journal of the South African Institute of Mining and Metallurgy, **110(8)**, pp. 437–446.

[74] Newman A & Kuchta M, 2007, *Using aggregation to optimize long-term production planning at an underground mine*, European Journal of Operational Research, **176(2)**, pp. 1205–1218.

[75] Newman A, Rubio E, Caro R, Weintraub A & Eurek K, 2010, *A review of operations research in mine planning*, Interfaces, **40(3)**, pp. 222–245.

[76] Orlowski S & Pióro M, 2012, *Complexity of column generation in network design with path-based survivability mechanisms*, Networks, **59(1)**, pp. 132–147.

[77] O'Sullivan D & Newman A, 2015, *Optimization-based heuristics for underground mine scheduling*, European Journal of Operational Research, **241(1)**, pp. 248–259.

[78] Padberg M & Rinaldi G, 1991, *A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems*, SIAM Review, **33**, pp. 60–100.

[79] Pritsker A, Watters L & Wolfe P, 1969, *Multi-project scheduling with limited resources: A zero-one programming approach*, Management Science, **16**, pp. 93–108.

[80] *PSPLIB — A project scheduling problem library*, 1996, [Online], [Cited July 2015], Available from http://www.om-db.wi.tum.de/psplib/main.html.

[81] Quilliot A & Toussaint H, 2013, *Flow models for project scheduling with transfer delays and financial constraints*, Studies in Computational Intelligence, **470**, pp. 135–154.

[82] Rahal D, Smith M, van Hout G & von Johannides A, 2003, *The use of mixed integer linear programming for long-term scheduling in block caving mines*, Proceedings of the 31st International APCOM Symposium, Cape Town, pp. 123–132.

[83] Rubio E & Diering E, 2004, *Block cave production planning using operation research tools*, Proceedings of the International Conference and Exibition on Mass Mining, Santiago, pp. 141–149.

[84] Sarin S & West-Hansen J, 2005, *The long-term mine production scheduling problem*, IIE Transactions, **37(2)**, pp. 109–121.

[85] Schrage L, 1970, *Solving resource-constrained network problems by implicit enumeration — non-preemptive case*, Operations Research, **18**, pp. 263–278.

[86] Schulze M & Zimmermann J, 2011, *Scheduling in the Context of Underground Mining*, pp. 611–616 in Hu B, Morasch K, Pickl S & Siegle M (Eds), *Operations Research Proceedings 2010*, Springer, Berlin.

[87] Schutt A, Feydy T, Stuckey P & Wallace M, 2009, *Why cumulative decomposition is not as bad as it sounds*, Lecture Notes in Computer Science, **5732**, pp. 746–761.

[88] Schutt A, Feydy T, Stuckey P & Wallace M, 2013, *Solving the resource-constrained project scheduling problem with generalized precedences by laxy clause generation*, Journal of Scheduling, **16(3)**, pp. 273–289.

[89] Sirdey R & Kerivin H, 2007, *A branch-and-cut algorithm for a resource-constrained scheduling problem*, RAIRO — Operations Research, **41**, pp. 235–251.

[90] Smith M, Sheppard I & Karunatillake G, 2003, *Using MIP for strategic life-of-mine planning of the lead/zinc stream at Mount Isa Mines*, Proceedings of the 31st International APCOM Symposium, Cape Town, pp. 465–474.

[91] Smith-Daniels D & Smith-Daniels V, 1987, *Maximizing the net present value of a project to materials and capital constraints*, Journal of Operations Management, **7**, pp. 33–45.

[92] Stinson J, Davis E & Khumawala B, 1978, *Multiple resource-constrained scheduling using branch and bound*, AIIE Transactions, **10**, pp. 252–259.

[93] Terblanche S & Bley A, 2015, *An improved formulation of the underground mine scheduling optimisation problem when considering selective mining*, ORiON, **31(1)**, pp. 184–197.

[94] Topal E, 2008, *Early start and late start algorithms to improve the solution time for long-term underground mine production scheduling*, The Journal of the South African Institute of Mining and Metallurgy, **108**, pp. 99–107.

[95] Turing A, 1937, *On computable numbers, with an application to the Entscheidungsproblem*, Proceedings of the London Mathematical Society, **2(42)**, pp. 230–265.

[96] Uetz M, 2001, *Algorithms for deterministic and stochastic scheduling*, PhD Thesis, Technische Universität, Berlin.

[97] Vanhoucke M, Demeulemeester E & Herroelen W, 2001, *On maximizing the net present value of a project under renewable resource constraints*, Management Science, **47**, pp. 1113–1121.

[98] Vossen TWM, Wood RK & Newman AM, 2016, *Hierarchical benders decomposition for open-pit mine block sequencing*, Operations Research, **64(4)**, pp. 771–793.

[99] Weaver P, 2006, *A brief history of scheduling*, [Online], [Cited July 2014], Available from `http://www.mosaicprojects.com.au/Resources%5C_Papers%5C_042.html`.

[100] Whittle J, 1998, *Four-X*, Whittle Programming Pty Ltd, Melbourne.

[101] Yang K, Talbot F & Patterson J, 1992, *Scheduling a project to maximize its net present value: An integer programming approach*, European Journal of Operational Research, **64**, pp. 188–198.

[102] Zapata J, Hodge B & Reklaitis G, 2008, *The multimode resource constrained multi-project scheduling problem*, AIChE Journal, **54(8)**, pp. 2101–2119.

[103] Zhu G, Bard J & Gang Y, 2006, *A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem*, INFORMS Journal on Computing, **18(3)**, pp. 377–390.